Universidade de Aveiro Departamento de
Electrónica, Telecomunicações e Informática,
2017

**João Pedro
Ferreira e Pereira**

Dissemination of contextual information for assisted driving

Universidade de Aveiro Departamento de Electrónica, Telecomunicações e Informática,
2017

João Pedro Ferreira e Pereira

Dissemination of contextual information for assisted driving

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requesitos necessários à obtenção do grau de Mestre em Engenharia Electrónica e Telecomunicações, realizada sob a orientação científica de André Zúquete, Professor do Departamento de Electrónica, Telecomunicações e Informática da Universidade de Aveiro, e do Dr. Lucas Guardalben, Investigador do Instituto de Telecomunicações - Aveiro.

o júri / the jury

presidente / president — **Professor Doutor António Guilherme Rocha Campos,**
Professor Auxiliar do Departamento de Electrónica, Telecomunicações e Informática da Universidade de Aveiro

vogais / examiners committee — **Professor Doutor Pedro Nuno Miranda de Sousa**
Professor Auxiliar da Universidade do Minho - Escola de Engenharia

Professor Doutor André Ventura da Cruz Marnoto Zúquete
Professor Auxiliar do Departamento de Electrónica, Telecomunicações e Informática da Universidade de Aveiro (Orientador)

agradecimentos

Em primeiro lugar agradeço à minha família, que sempre estiveram presentes quando mais precisei e que sem eles nunca teria aqui chegado. Agradeço a todos os meus amigos e colegas que tive o prazer de conheçer durante este percurso, que não só contribuíram para a minha formação académica mas que essencialmente me ajudaram a ser uma pessoa melhor.

Gostava de agradecer também ao nosso grupo de investigação, aos colegas pela entreajuda e acolhimento, e aos investigadores pelo incalculável conhecimento e espírito crítico. Agradeço também aos professores André Zúquete e Lucas Guardalben pela orientação e pela ajuda prestada nesta dissertação. E por último, um especial agradecimento à professora Susana Sargento pela oportunidade de colaborar neste grupo e ao professor Miguel Díaz-Cacho pela ajuda durante o desenvolvimento deste trabalho.

Resumo

Os sistemas de condução assistida podem ser utilizados para melhorar a segurança rodoviária e automóvel, reduzir a fadiga da condução e proporcionar uma experiência de condução mais eficiente. Uma parte importante desses sistemas é a comunicação entre veículos e comunicação veículo-infraestrutura. Este trabalho propõe mecanismos que permitem o suporte à condução, explorando a rede de veicular para fornecer informações sobre a vizinhança do condutor.

A rede é composta por veículos, sinais de trânsito e estações fixas ao longo da estrada. Cada carro está equipado com uma câmera de gravação, um receptor GPS, bem como módulos de comunicação, como WiFi, WAVE e 3G/4G, permitindo a troca de dados entre os vários nós. Os dados trocados são compostos por dados posicionais de veículos vizinhos, informações sensoriais de sinais de trânsito e imagens de vídeo provenientes de outros veículos. Esses dados são usados para facilitar a tomada de decisões, mas também podem fornecer uma visão geral da densidade de tráfego na vizinhança. Os sinais de trânsito transmitem a sua posição e, no caso de serem dinâmicos (como semáforos), o seu estado actual também é transmitido. As estações fixas estão equipadas com vários sensores e são usadas para fornecer dados ambientais.
O condutor pode aceder a todos os dados recolhidos através de informações visuais, num ecrã que contém um mapa da sua redondeza junto com a informação disponvel dos nós vizinhos.

O sistema proposto é avaliado através de testes reais em dois cenários distintos: urbano e auto-estrada. Os resultados mostram que o atraso da comunicação é maior no cenário da auto-estrada, principalmente devido às maiores distâncias entre os veículos e às velocidades mais elevadas. No entanto, resultados promissores em relação ao atraso máximo e ao número médio de retransmissões prevêm contribuições importantes para serviços futuros de condução assistida em geral, e assistência de ultrapassagem de veículos, em particular.

Abstract

Driver assistance systems can be used to improve road and car safety, reduce driving fatigue and provide a more efficient driving experience. An important part of these systems is the communication between vehicles, and vehicle-to-infrastructure communication. This work presents mechanisms enabling driving support, exploring the vehicular network to provide information about the drivers neighborhood.

The network is composed by vehicles, traffic signals and fixed stations along the road. Each car is equipped with a recording camera, a GPS receiver, as well as communication modules such as WiFi, WAVE and 3G/4G, allowing the exchange of data between the various nodes. The data exchanged is composed by positional data of neighboring vehicles, sensor information from traffic signals and video images incoming from other vehicles. This data is used to facilitate the driver in decision making, but can also provide an overview of the traffic density in the neighborhood. The traffic signals broadcast their position and if they are dynamic (such as traffic lights), their status is also transmitted. The fixed stations are equipped with numerous sensors and are used to provide environmental data.

The driver can access all the collected data via visual information, on a display screen that contains a map of the neighborhood along with the information available of the nearby nodes.

The proposed system is evaluated through real vehicular experiments in two distinct scenarios: urban and highway. The results show that the communication delay is higher in the highway scenario, mainly due to the distance between vehicles and travelling speeds. However, promising results regarding the maximum delay and the average number of retransmissions foresee important inputs for future services of assisted-driving, in general, and car-overtaking assistance, in particular.

Contents

Contents i

List of Figures v

List of Tables vii

Acronyms ix

1 Introduction 1
- 1.1 Context and Motivation . . . 1
- 1.2 Objectives . . . 1
- 1.3 Contributions . . . 2
- 1.4 Acknowledgements . . . 3
- 1.5 Document Structure . . . 3

2 State of the art 5
- 2.1 Introduction . . . 5
- 2.2 Vehicular Networks . . . 5
  - 2.2.1 Features . . . 5
  - 2.2.2 Challenges . . . 6
  - 2.2.3 Architecture . . . 7
  - 2.2.4 Data Dissemination . . . 8
  - 2.2.5 WAVE Protocol . . . 9
  - 2.2.6 Applications and Services . . . 11
- 2.3 Driver Assistance Systems . . . 11
  - 2.3.1 Overview . . . 12
  - 2.3.2 Collision Avoidance . . . 13
  - 2.3.3 Vision-based Detection . . . 13
  - 2.3.4 Traffic Signal Detection . . . 14
  - 2.3.5 Vehicle-to-Vehicle (V2V) Video Transmission . . . 14
- 2.4 Summary . . . 15

3 Proposed Architecture 17
- 3.1 Introduction . . . 17
- 3.2 Architecture Overview . . . 17
- 3.3 System Elements . . . 18
- 3.4 Road Traffic Awareness . . . 21

3.5 Traffic Signals Sensing . . . 22
3.6 User Feedback . . . 24
3.6.1 Connection Setup and Control . . . 25
3.6.2 Data Selection . . . 25
3.7 Vehicle-to-Vehicle Video Transmission . . . 29
3.8 Summary . . . 31

4 Integration and Implementation **33**
4.1 Introduction . . . 33
4.2 Overall Integration . . . 33
4.2.1 On Board Unit . . . 34
4.2.2 Display Screen . . . 35
4.3 GPS Position Collection . . . 36
4.3.1 Beacon Format . . . 37
4.3.2 Node Status Information Table . . . 38
4.3.3 Position Packet Format . . . 39
4.3.4 Process Overview . . . 39
4.4 Sensory Data Collection . . . 40
4.4.1 Traffic Signal Packet Format . . . 40
4.4.2 Network Scanning and Connection . . . 42
4.4.3 Process Overview . . . 43
ESP8266 . . . 43
On Board Unit . . . 43
4.5 Data Presentation . . . 45
4.5.1 Overview . . . 45
4.5.2 Communication Channel and Data Storage . . . 46
4.5.3 Data Fetching and Storage . . . 47
4.5.4 Map Update and Offline Maps Data . . . 47
4.6 Video Codification and Transmission . . . 48
4.6.1 Video Recording . . . 48
4.6.2 Encoding . . . 50
4.6.3 Data Forwarding and Visualization . . . 51
4.7 Summary . . . 51

5 Evaluation **53**
5.1 Introduction . . . 53
5.2 Data Selection and Display . . . 53
5.3 Vehicle-to-Vehicle Video Transmission . . . 55
5.3.1 Urban Scenario . . . 58
5.3.2 Highway Scenario . . . 61
5.3.3 Considerations . . . 63
5.4 Summary . . . 64

6 Conclusion and Future Work **67**
6.1 Conclusion . . . 67
6.2 Future Work . . . 68

Bibliography **69**

List of Figures

2.1 Vehicle Ad-hoc NETwork (VANET) applications (from [1]). 6
2.2 VANET Communication Domains (from [8]). 8
2.3 Data dissemination strategies (from [31]). a) depicts the typical case of a single-hop; b) represents a multi-hop dissemination. 10
2.4 Wireless Access in Vehicular Environments (WAVE) protocol stack (from [27]). 11
2.5 Potential evolution for automated cooperative driving (from [5]). 12
2.6 Different levels of vision for semantic interpretation (from [39]). 14
2.7 Different signal detection scenarios (from [29]). a) a fully automated driving system should detect all the traffic signals; b) with a driver on the vehicle, only the most important should be highlighted; c) inform a driver when he is found to be inattentive to the road. 15

3.1 Overall system architecture. 18
3.2 On Board Unit. 19
3.3 ESP8266. 20
3.4 GoPro Hero 4 Black edition. 21
3.5 Each On Board Unit (OBU) broadcasts it's GPS information 22
3.6 Road Traffic Awareness process 22
3.7 Architecture for the traffic signal sensing. 23
3.8 Sensory data flow. 24
3.9 Packet flow of the OBU and display screen. 25
3.10 Suggested Presentation. 26
3.11 Movement Detection. 27
3.12 Suggested representation taking into account the vehicles movement. 27
3.13 Suggested representation with both suggested filters. 28
3.14 Traffic signal heading visualization. 29
3.15 Typical use-case. 30
3.16 Software architecture of the video transmission. 30
3.17 Software architecture of the video reception. 30
3.18 Video transmission overview. 31

4.1 On Board Unit software architecture. 34
4.2 Display screen software architecture. 35
4.3 Node Status Information frame. 37
4.4 Position packet format 39
4.5 Position collection process flow 41

4.6 Traffic signal packet format 42
4.7 Process of the traffic signal 44
4.8 Sensory data collection flow 45
4.9 Overview of the display screen state machine. 46
4.10 Communication Channel and Data Storage state diagrams. 47
4.11 Data Fetching and Data Storage state diagrams. 48
4.12 Map Update and Offline Maps Data state diagrams. 49

5.1 Movement display: calibration of the τ parameter. 54
5.2 Example of the relative position display. 55
5.3 Example of the traffic light display. 56
5.4 V2V video stream setup. 57
5.5 Urban scenario - Comparison between packed delay and speed of the receiving vehicle. 59
5.6 Urban scenario - Comparison between packed delay and speed of the transmitting vehicle. 59
5.7 Urban scenario - Comparison between packed delay and quality of the IEEE 802.11p link. 60
5.8 Highway scenario - Comparison between packed delay and speed of the receiving vehicle. 61
5.9 Highway scenario - Comparison between packed delay and speed of the transmitting vehicle. 62
5.10 Highway scenario - Comparison between packed delay and quality of the IEEE 802.11p link. 62

List of Tables

4.1 OBU's SSID and Addressing 42
4.2 Internet Protocol (IP) addressing and port mapping 51

5.1 Overall average results for the urban scenario. 60
5.2 Overall average results for the highway scenario. 63
5.3 Video transmission availability 64

Acronyms

ACC	Adaptive Cruise Control
AP	Access Point
API	Application Programming Interface
ARM	Advanced RISC Machine
AU	Application Unit
BSS	Basic Service Set
C2C-CC	Car-to-Car Communication Consortium
C2C	Car-to-Car
CCA	Cooperative Collision Avoidance
CWS	Collision Warning System
DAS	Driver Assistance Systems
DHCP	Dynamic Host Control Protocol
DSRC	Dedicated Short Range Communication
FPS	Frames Per Second
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
GPRS	General Packet Radio Service
GSM	Global System for Mobile Communications
HSDPA	High Speed Packet Access
HTTP	Hypertext Transfer Protocol
I2V	Infrastructure-to-Vehicle
IEEE	Institute of Electrical and Electronics Engineers
IoT	Internet of Things

IP	Internet Protocol
LAN	Local Area Network
LTE	Long-Term Evolution
MAC	Media Access Control
MANET	Mobile Ad hoc NETworks
MJPEG	Motion JPEG
NSI	Node Status Information
NTP	Network Time Protocol
OBU	On Board Unit
OCR	Optical Character Recognition
POV	Points Of View
PSID	Provider Service ID
RSSI	Received Signal Strength Indicator
RSU	Road Side Units
SBC	Single-Board Computer
SPI	Serial Peripheral Interface
SSID	Service Set Identifier
TCP	Transmission Control Protocol
TDMA	Time Division Multiple Access
TTL	Time To Live
UART	Universal Asynchronous Receiver-Transmitter
UDP	User Datagram Protocol
UMTS	Universal Mobile Telecommunications System
URL	Uniform Resource Locator
UWM	Ultra-wideband
V2V	Vehicle-to-Vehicle
V2I	Vehicle-to-Infrastructure
VANET	Vehicle Ad-hoc NETwork
VSA	Vendor Specific Action
WAVE	Wireless Access in Vehicular Environments
WSUB	Wireless USB

Chapter 1

Introduction

This chapter describes the context and motivations for this dissertation, followed by its objectives. The contributions of this work and acknowledgements are also presented, along with a brief description of the structure of the document.

1.1 Context and Motivation

VANETs constitute a major research area nowadays, driven by the necessity of the public in general to be always connected to the Internet. It is expected that every vehicle can take part of such networks, opening a wide variety of both entertainment and safety applications. For entertainment, all passengers of a vehicle can have access to the Internet at all times, enabling the possibility to use video on-demand, online games, browsing, etc.; on the other hand, VANETs also provide a means to advertise local businesses or tourist attractions. The communications provided by these networks are essential to a vast number of safety applications, as every vehicle is able to send emergency data to all the surrounding vehicles to, for example, prevent accidents or to reduce traffic congestion.

Both private and public transport systems are used by many people on a daily basis. With the increase in the usage of private transports, the number of fatalities also increases. Although there has been an improvement in road safety in the last decades, road accidents are still one of the leading causes of death worldwide [49]. As such, road safety remains a major concern not only for the general public, but also for the automotive industry. One of the solutions that car manufacturers propose is the development of mechanisms capable of offering assisted driving mechanisms or even totally autonomous transport systems.

A significant part of Driver Assistance Systems (DAS) requires the ability to connect and exchange data between vehicles, both to enable the collection of more environmental information and to improve decisions. The main motivation for this dissertation is, therefore, the creation of a platform capable of assisting a driver in decision making, using the V2V and Vehicle-to-Infrastructure (V2I) communications provided by a VANET.

1.2 Objectives

The main objective of this work is to provide a platform capable of collecting external information about a vehicle's neighborhood, as well as displaying the retrieved data to a driver. As such, the objectives for this dissertation are to:

- Study the main characteristics of VANETs, as well as the developments and existing testbeds for DAS.
- Create a system capable of collecting positional information about neighboring vehicles.
- Develop an application adequate to collect sensory data incoming from nearby traffic signals.
- Set the grounds to the possibility of improving the system with the support of other types of sensory data.
- Design and implement a feedback system for the driver, based on an Android application.
- Implement a video transmission service between multiple vehicles using V2V communication.
- Evaluate the overall functional behavior of the implemented strategies by performing tests on a real environment.

1.3 Contributions

This work contributed in the following points:

- Creation of a platform for development, implementation, and testing of assisted driving related applications.
- Implementation of several services capable of collecting and displaying contextual information regarding the surrounding environment of a vehicle.
- Evaluation in both an urban and highway environments of vehicle-to-vehicle video transmissions using Institute of Electrical and Electronics Engineers (IEEE) 802.11p, and evaluation of neighboring information dissemination in a real platform.

Part of the work developed during this dissertation was accepted and presented on the 3rd International Conference on Event-Based Control, Communication and Signal Processing in May $24-26$ of 2017, entitled Send-on-Delta sampling strategies for vehicle position tracking.

The work was presented under the name of "Message Dissemination Mechanisms for Driving Support Scenarios" on the 23rd Rede Tem'atica de Comunicações Móveis Seminar on July 18.

Under the same name, it was accepted and presented in INFORUM 2017, from $12-13$ of October, in the form of a communication, an oral presentation and a poster, where it received an award for "best communication poster".

A submission entitled "Vehicular Context-Aware Dissemination for Assisted Driving" was targeted to IEEE Vehicular Technology Conference (VTC) Spring 2018 and is currently under evaluation.

1.4 Acknowledgements

This work was supported in part by National Funds through FCT - Fundação para a Ciência e a Tecnologia under the project UID/EEA/50008/2013, and in part by the Europe Union project EU H2020-ICT 732497 5GinFIRE.

1.5 Document Structure

This document is organized in the following manner:

Chapter 2 - State of the Art : Provides an overview of the state of the art for vehicular networks and driver assistance systems.

Chapter 3 - Proposed Architecture : Presents the overall architecture proposed, with a description of all the involved elements and services created.

Chapter 4 - Integration and Implementation : Explains the integration and implementation details of the proposed services.

Chapter 5 - Evaluation : Discusses the scenarios tested and the results obtained.

Chapter 6 - Conclusions and Future Work : Contains the conclusion of this dissertation and proposes some improvements for future research.

Chapter 2

State of the art

2.1 Introduction

This chapter presents the main concepts needed to understand the work of this dissertation along with its related work on the fundamental topics. It is organized as follows:

- Section 2.2 provides an overview and definition of a VANET. This includes a discussion about their characteristics, challenges, overall architecture, data dissemination methods, the WAVE protocol and general applications and services.
- Section 2.3 provides some related work on DAS, as well as an insight on some of the developments and currently available testbeds.

2.2 Vehicular Networks

A VANET consists on an ad hoc network formed by moving vehicles and fixed stations, equipped with devices that provide wireless communication capabilities to enable the transmission of data between themselves, as shown in Figure 2.1. The devices within the vehicles are denominated OBU and correspond to the mobile nodes of the network. In order to provide Internet access to these nodes, fixed stations are installed along the roads, called Road Side Units (RSU), which are connected to the infrastructure. As such, two types of communication are possible: V2V communication when the OBUs communicate with each other, or V2I when an OBU communicates directly with the infrastructure.

2.2.1 Features

When compared against other Mobile Ad hoc NETworks (MANET), a VANET has a unique set of characteristics [1][27][38][34], including:

Sufficient energy and resources The vehicle's engine provides continuous power that can be used to support the OBU or other computational devices present on the vehicle.

Predicted mobility Differently from other MANETs, the mobile nodes' behaviour in a VANET is very predictable as vehicles are limited to travel along roads and follow a set of rules (speed limits, stop signs, traffic lights, etc). Furthermore, every vehicle can be

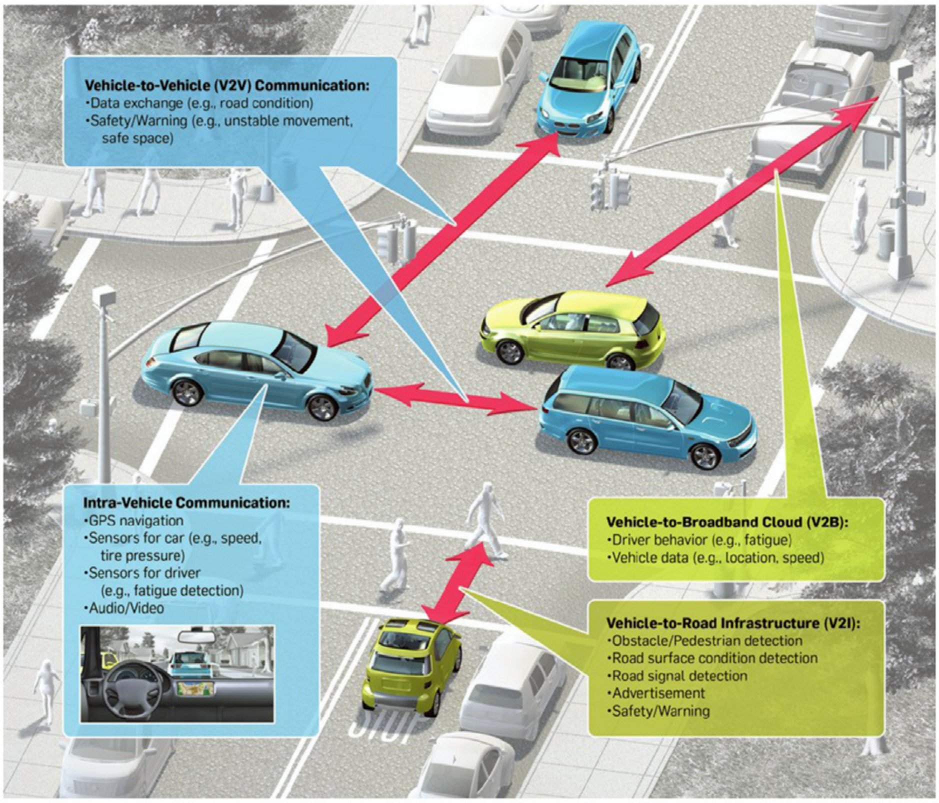

Figure 2.1: VANET applications (from [1]).

easily equipped with a Global Positioning System (GPS) receiver, allowing the tracking and prediction of their movement.

Dynamic network topology As vehicles are the communication nodes and are highly mobile, the topology of the network is constantly changing, resulting in a very dynamic network.

2.2.2 Challenges

The main prerequisites and characteristics of a VANET result in a variety of challenges [27][17][55][53], such as:

Security and privacy One of the main challenges of a VANET is to maintain a reasonable balance between the security and privacy of the users. The receiver wants to ensure that the information received can be trusted. However, the viability of the information may violate the sender's privacy requirements.

Network fragmentation The high mobility of the nodes that integrate the VANET can lead to constant changes in the network topology and very often to breaks on the link between the various nodes. The solutions typically proposed for the WAVE connection disruption involve using cellular links, which is not ideal as it results in higher cost for the network exploitation.

Signal degradation In an environment with multiple obstacles, such as city buildings and neighboring vehicles, the signal transmitted between vehicles may be attenuated to a level where it is unable to reach the target destinations.

Accurate positioning A certain accuracy in the measurements of the position of the vehicles is necessary for routing protocols based on the position of the nodes and also for certain safety applications. Most OBUs come equipped with a Global Navigation Satellite System (GNSS) receiver, which is relatively accurate on open areas where there is a direct line of sight to several satellites. Still, signal blockages may occur in situations such as tunnels or generally on downtown areas. Overall, the accurate measurement of the position of the various nodes can help improving the performance of such routing protocols or applications. As a result, substantial improvements are still necessary both in the accuracy and the availability of the position receivers.

Latency constraints Some safety applications may require a low communication delay to function properly. The time lapse of the exchange of information needs to be minimized and bounded, since an unpredictable delay may result on the incorrect or inaccurate behavior of certain applications, such as Cooperative Collision Avoidance (CCA).

2.2.3 Architecture

As stated by Al-Sultan et al. [1] a VANET's architecture is composed by three components:

OBU Installed in the vehicles, the OBU makes use of the WAVE technology to establish connections and exchanged data with nearby vehicles and fixed stations. These devices can also be equipped with other communication technologies such as IEEE 802.11 a/b/g/n or cellular interfaces. The main roles of the OBUs include radio access, ad hoc routing, data relay and IP mobility.

Application Unit (AU) Alongside the OBU, the AU is an equipment inside the vehicles. The AU can be a specific device to run safety applications or can represent any device demanding Internet access via the OBU. As such, the AU and the OBU can also integrate a single component, where the only difference is the set of functions executed by each one.

RSU The RSU is an equipment that is generally positioned alongside the roads or on strategic places, such as traffic lights or near parking lots. In order to be able to communicate with the OBUs, these devices are also equipped with WAVE. Furthermore, other technologies can be used to connect to the infrastructure. The main functions of the RSU are to expand the communication range of the network by relaying the information between other RSUs and OBUs, act as an information source for safety applications using Infrastructure-to-Vehicle (I2V) communication and also to provide Internet access to the OBUs.

The Car-to-Car Communication Consortium (C2C-CC)[8] proposed an overview of the architecture for a Car-to-Car (C2C) communication system by defining three different communication domains, as portrayed in Figure 2.2. They are defined as follows:

In-Vehicle domain This domain fundamentally is a Local Area Network (LAN) containing one OBU and possibly multiple AUs. The connection between the AUs and the OBU is typically wired, but can also be a wireless link using Bluetooth, Wireless USB (WSUB) or Ultra-wideband (UWM).

Ad hoc domain The ad hoc domain is formed by the vehicles containing OBUs allowing the communication between all the mobile nodes. The vehicles can also establish connections with RSUs located along the road.

Infrastructure domain Since an RSU has access to outer networks, it can be used, for example, to provide Internet access to surrounding OBUs. If there is are no RSUs available nearby, the OBUs can also access the Internet using cellular links such as Global System for Mobile Communications (GSM), General Packet Radio Service (GPRS), Universal Mobile Telecommunications System (UMTS), High Speed Packet Access (HSDPA), or Long-Term Evolution (LTE).

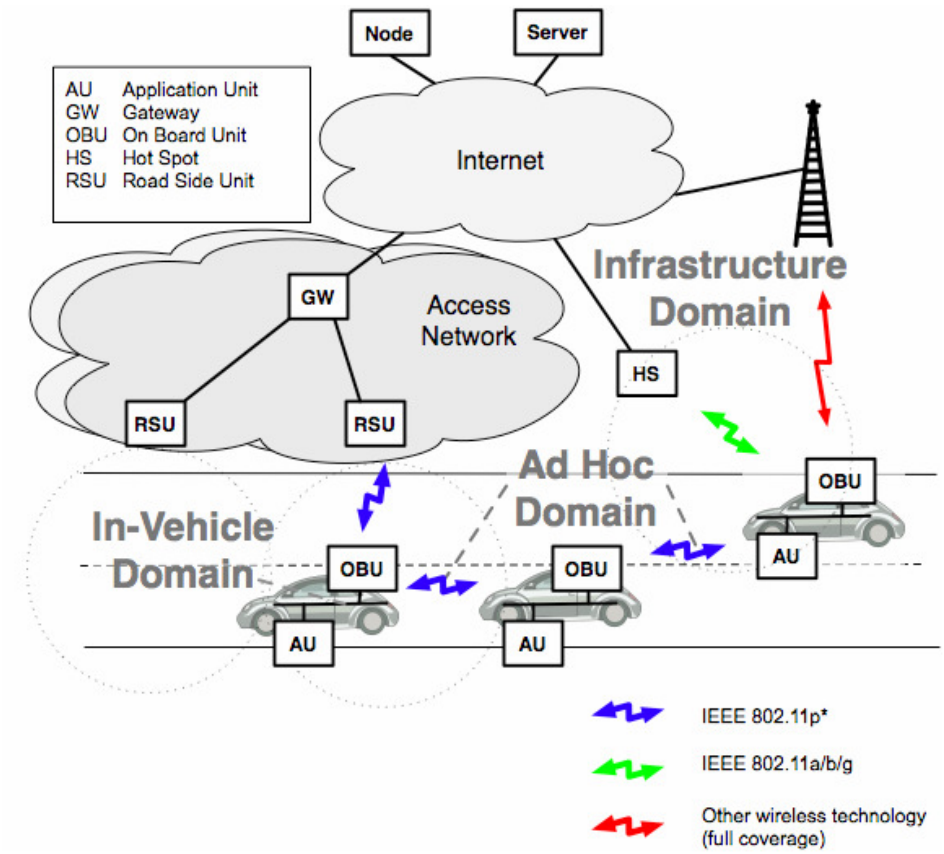

Figure 2.2: VANET Communication Domains (from [8]).

2.2.4 Data Dissemination

Generally, all the applications running on a VANET requires the mobile nodes to continually send data. Hence it is required to define a set of strategies for the way the information can be exchanged. The data dissemination strategies are depicted on Figure 2.3, and can be assorted as one of two types:

Single-hop Usually a broadcast on the Media Access Control (MAC) layer. In the case of the Figure 2.3, the vehicle A sends data to all available neighbors. The vehicle B is out range, and as such does not received the information from A.

Multi-hop This strategy is based on the constant forwarding of the data until it reaches the destination, using intermediary mobile nodes to relay the information. For the specific case of the Figure 2.3, using this strategy provides a wider communication range, allowing the vehicle A to contact vehicle B.

Depending on the number of destinations, the dissemination can be done in the following manners:

Unicast Only one sender and one receiver. Typically used for entertainment applications such as video streaming, games and general-purpose Internet access.

Multicast With one sender and one or more receivers. Usually used for safety-related applications that aim to send information to a specific region or group of vehicles.

Broadcast With one sender and one or more receivers. Used for control plane protocols, routing, etc. Also used for safety applications, to disseminate the data to all the surrounding mobile nodes.

2.2.5 WAVE Protocol

The communication infrastructure needed to achieve V2V communication and to collect the environment status of the road is still under research and development. Suggested 5G mobile communications infrastructure [9] includes specific challenges to improve quantitatively certain indicators, such as minimum bandwidth, delays, coverage area and quality of service with respect to the actual 3G/4G infrastructure to be used in V2V communications. Nevertheless, several specific communication technologies are already available to develop applications that involve the complexity of the V2V communication, most notably the IEEE 802.11p [30]. The IEEE 802.11p technology has been developed for dynamic, vehicular environments: it is able to provide a communication range of up to 900m in line-of-sight, which is very important to reach vehicles in a road environment, as well as connections establishment times in the order of $10 - 20$ ms, which enable very small opportunities of communication.

To address the specific characteristics of VANETs, IEEE developed the WAVE protocol stack as shown in Figure 2.4. The WAVE protocol stack is composed by the IEEE 802.11p[22] and IEEE 1609.x[24] standards.

IEEE 802.11p

IEEE standard defines a set of enhancements to the 802.11a standard to provide a base standard capable of addressing the main characteristics of a VANET. According to Jiang and Delgrossi, a set of modifications were deployed in the physical layer, most notably, the frequency band changed from 5.0 GHz to 5.9 GHz and the use of channels with 10 MHz

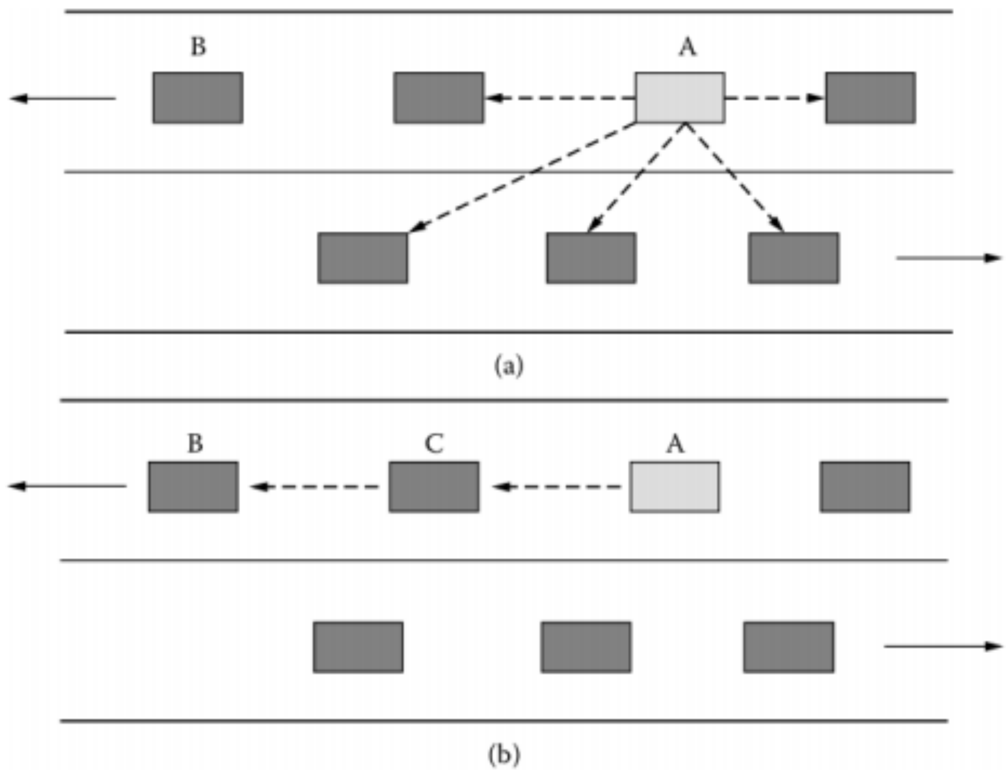

Figure 2.3: Data dissemination strategies (from [31]). a) depicts the typical case of a single-hop; b) represents a multi-hop dissemination.

instead of the usual 20 MHz. This standard also introduces some modifications in the MAC layer, specifically simplification of the Basic Service Set (BSS) operation, by allowing the possibility to communicate outside of a BSS context.

IEEE 1609.x

The IEEE 1609 family is composed by four leading standards. Generally, it is responsible for the control and management of services that the MAC layer provides. They are described as follows:

IEEE 1609.1 [21] : Provides a resource manager for WAVE by specifying the interaction between the OBU present in the vehicle and other remote computational resources.

IEEE 1609.2 [19] : This standard defines the WAVE security messages and mechanisms performed to ensure secure communication, with secure data and service advertisements.

IEEE 1609.3 [23] : This standard defines network and transport layer services, along with addressing and routing services.

IEEE 1609.4 [20] : This standard provides enhancements to the IEEE 802.11 MAC to support WAVE operation and services that require multi-channel operations.

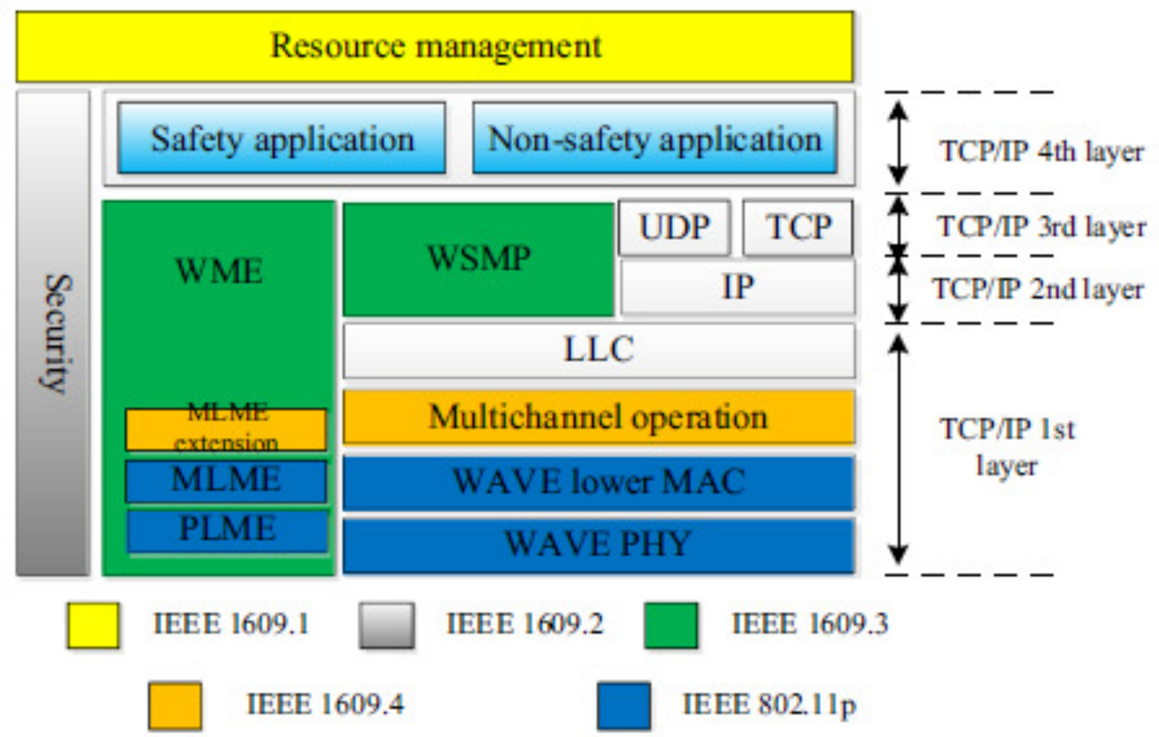

Figure 2.4: WAVE protocol stack (from [27]).

2.2.6 Applications and Services

The combination of V2V and V2I communications result in a wide variety of applications and services to drivers and passengers [38][17][53][51], including:

Safety The V2V communications enable the development of applications that can significantly improve the road and public safety. Some of the examples under this category include CCA, sign extension and vehicle diagnosis.

Traffic monitoring These applications provide information of the surrounding environment including the position or density of nearby vehicles. The availability of the traffic information of a certain region can lead to faster trips and reduce overall traffic jams.

Entertainment The objective of this type of applications is generally to improve the drivers and the passengers comfort. The collection of local information, such as tourist attractions, restaurants or petrol stations, are among the most common. Online games, video on-demand and other typical applications that require Internet access also fall under this category.

2.3 Driver Assistance Systems

DAS are one of VANET's most important targets as an application, focusing not only on safety or crash reduction but also on improving the overall driving experience. This section provides an overview of some of the developments, research lines and current available experiments for DAS.

2.3.1 Overview

According to Bengler et al.[5] early developments integrated sensors within the vehicle, measuring metrics such as velocity or acceleration. Further advances led to the use of external sensors such as GNSS receivers, yielding information of the position of the vehicle and about the road ahead. Navigation systems can be important for the reduction of the risk of car accidents as they can aid in orientation, providing a way to reduce a driver's workload[44]. The following milestone of these systems was parking assistance expanding initially from parallel to perpendicular parking, using technologies such as ultrasounds and 360^o cameras. Adaptive Cruise Control (ACC) was also an important achievement, using electronic brakes and radars to provide some drive control. Collision prevention systems constitute one of the more recent advances and research topics. Particularly, CCA for on-road vehicles is one of the main use-case examples in the development of new vehicular control systems and mobile network infrastructures, as recognized in the recently published 5G white paper for automotive and mobility [9]. Figure 2.5 shows some of the developments and general future expectations of evolution for cooperative driving.

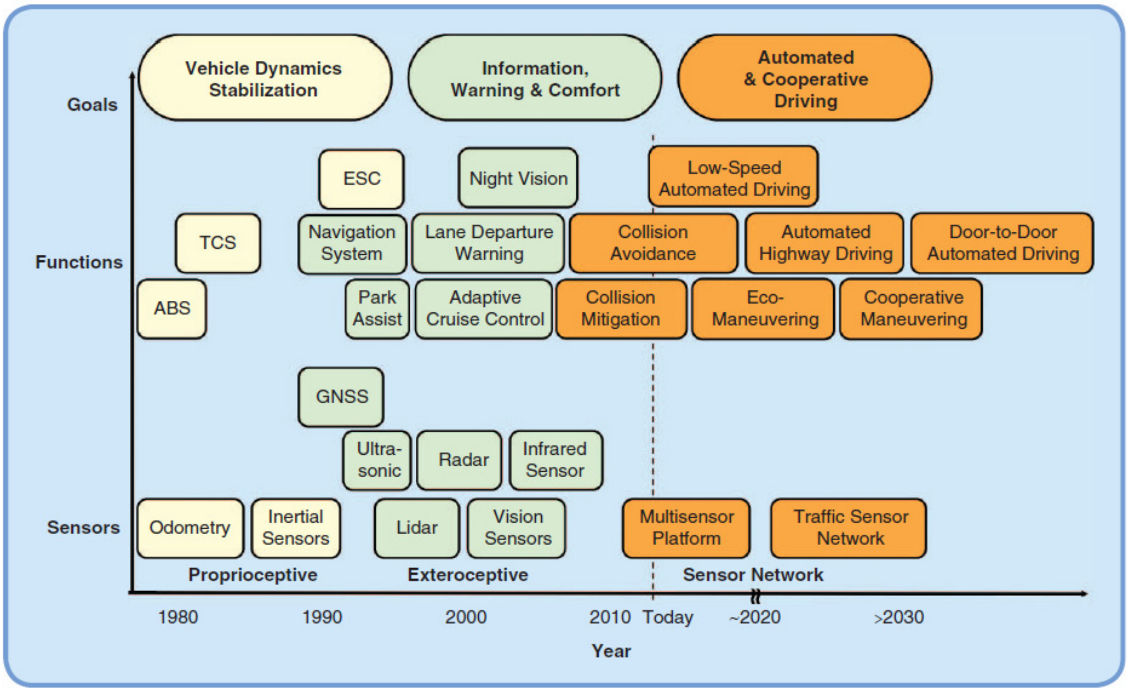

Figure 2.5: Potential evolution for automated cooperative driving (from [5]).

More recently, some of these research lines were focused on improving the position accuracy effectiveness to overcome the error measurements of the GNSS sensors in urban areas, mainly because of the multi-path effects and the absence of sufficient satellite signals in GNSS receivers. They use cooperative positioning techniques [2] and road-feature measurements by fusing the data received by the on-board installed GNSS sensor with the data received by other connected vehicles and geo-referenced lane boundaries [6]. Nevertheless, we consider that the future GNSS sensors will have enough accuracy to provide reliability for car driving assistance [50]. In this case, the use of information provided by other vehicles and environmental sensors via the V2V communication technologies deeply improves the decision taken by the driver and the driving controller of the vehicle.

The research lines in DAS are focused on the vehicular control technologies, and therefore, on the on-board installed sensors, controllers and actuators [32]. Nevertheless, in the recent years, the use of communication links between vehicles, and between them and the cloud network infrastructure to share sensor, actuator and control data is becoming more common [6, 54, 2].

2.3.2 Collision Avoidance

Collision avoidance is currently one of the most interesting research areas in the DAS field. The main research and implementation lines are centered on the on-board installed sensors by merging the different sensor data to calculate collision risks or estimate overtaking decisions [41, 13, 18, 36]. Several cooperative applications, such as Collision Warning System (CWS), require that the self generated control data (sensors, controllers and actuators data) is received by all the surrounding vehicles, whether they are integrated in a VANET or not. From [32] and its associated bibliography, the most important sensors involved in an assisted driving vehicle are the pan-tilt-zoom camera, the photonic mixer device, the laser scanner, the short-range radar, the long range radar, the ultrasonic sensors (sonars), the fixed camera and the differential GPS.

Other research lines for collision avoidance are focused on CCA, creating message protocols [52], establishing the base requirements for cooperative CWS [42] and designing decision-making modules using VANETs also for cooperative CWS [54]. Tatchikou et al.[43] proposed mechanisms to avoid chain-collisions by generating and delivering collision warning messages using Dedicated Short Range Communication (DSRC) and V2V communication when an emergency event occurs. Hafner et al.[15] on the other hand, proposed a set of rules and algorithms using V2V communication to create a system capable of detecting possible collisions before they happen, and automatically actuate to try to avoid vehicle crashes.

2.3.3 Vision-based Detection

Another important research topic of DAS is the on-road vision-based detection of vehicles, particularly over the past decades [41]. According to Sivaraman and Trivedi[39], progresses in the computational systems, camera technologies and sensor fusion with vision, enabled the extensive study of vision-based systems with reliable recognition capabilities. Also according to Sivaraman and Trivedi[39], algorithmic advances have made possible the high level of semantic interpretation of the road environment. Figure 2.6 portraits three different levels of vision for semantic interpretation. McCall and Trivedi[28] suggested a system capable of analyzing video images to recognizing the road lanes, an important part that can be further used on assisted or even automated driving systems. Wen et al.[48] propose solutions to speed up the machine learning process with the design of a incremental learning algorithm, that can improve the classification performance significantly when compared to other state of the art methods.

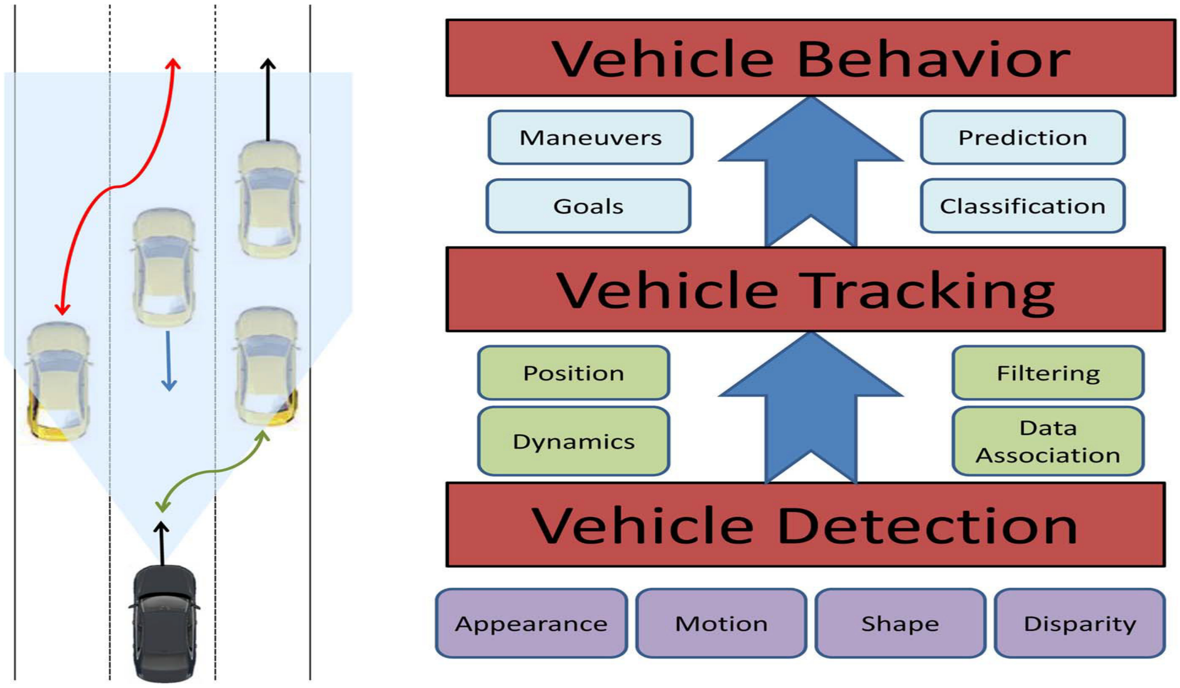

Figure 2.6: Different levels of vision for semantic interpretation (from [39]).

2.3.4 Traffic Signal Detection

According to Mgelmose et al.[33], traffic signal detection is crucial to intelligent vehicles and in general to DAS. Traffic signal information is usually processed by the vehicle using artificial vision procedures. Image sensors have to focus to the traffic signal, to select the relevant information and to interpret its meaning by using Optical Character Recognition (OCR) and other technologies [33]. Mogelmose et al.[29] explained the different possible detection scenarios depending on the application, as shown in Figure 2.7. Figure 2.7 a) depicts how a fully automated driving system should consider the traffic signals data, that is, it should consider all the signs present. Figure 2.7 b) considers that there is a driver present on the vehicle, and as such, even if the system is able to detect all the traffic signs, only the most significant signs are displayed to avoid confusion. The scenario in Figure 2.7 c) is more complex. In this case, it also takes into the account the attentiveness of the driver. When the driver is found to be inattentive to the road, a feedback system activates to inform him of the nearby traffic signs. A possibility for this feedback is using a heads-up display [11].

2.3.5 V2V Video Transmission

Real-time video transmission in vehicular networks still have room for improvements, specially when the concern relies on real-time fast and reliable mechanisms for video dissemination [7], [35]. A lot of recent research efforts have been dedicated to the efficient video-streaming in VANETs [40], and simulation models are mostly used as proof-of-concept for quality video assessment covering several general features of the IEEE 802.11p communication networks [46, 37, 45, 26].

The work in [16] includes a comparison between LTE and IEEE 802.11p standards, where

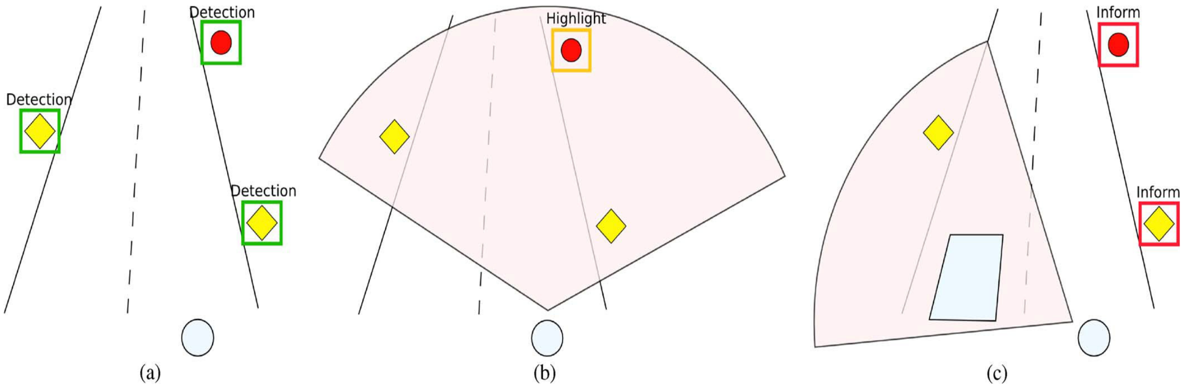

Figure 2.7: Different signal detection scenarios (from [29]). a) a fully automated driving system should detect all the traffic signals; b) with a driver on the vehicle, only the most important should be highlighted; c) inform a driver when he is found to be inattentive to the road.

IEEE 802.11p offers an acceptable performance for sparse network topologies with limited mobility support, and LTE meets most of the application requirements in terms of reliability, scalability, and mobility support. However, it is challenging to obtain stringent delay requirements in the presence of higher cellular network traffic load. This result proves that IEEE 802.11p is still a very suitable standard for inter-vehicules real-time video imaging as one of the focuses of this paper. Therefore, just a few works considered real-world experiments on video measurements over IEEE 802.11p as presented in [47, 4]. Gomes et al.[14] proposed a system that makes use of multiple technologies such as cellular and V2V communications, as well as laser holographic projections, to create a video streaming process between two vehicles that is capable of providing some insight on car overtaking situations when a lot of vision is taken way, as is the case of a truck or a bus in front. Vinel et al. proposed a real-time scalable video codec for the video information, and performed real-world measurements using the off-the-shelf Componentality FlexRoad DSRC equipment. However, it considered only experiments in urban scenarios close to the university campus in Hervanta, a suburb of Tampere, Finland.

2.4 Summary

This chapter presented the overview of the fundamental concepts of this dissertation, starting with the presentation of the main characteristics and concepts of a VANET. Then, a description of the major developments for DAS was presented, along with some insight on the main research areas of this topic. In terms of related work, some current testbeds and implementations were also discussed.

Chapter 3

Proposed Architecture

3.1 Introduction

This chapter presents the proposed architecture with its requirements. Furthermore, it also presents the technologies and mechanisms involved in the exchange of information between all the nodes that compose the network, along with the services developed to gather and interpret the collected data to provide feedback to a driver.

This chapter is outlined in the following way:

- Section 3.2: This section describes the overall architecture of the network along with its requirements.
- Section 3.3: The entities involved in this system are presented.
- Section 3.4: Describes how the information regarding vehicles is collected.
- Section 3.5: The service that gathers information of the traffic signals is presented.
- Section 3.6: Provides an overview of the mechanisms proposed to send and display the collected data in a visual way.
- Section 3.7: Details the structure of the Vehicle-to-Vehicle video transmission system.

3.2 Architecture Overview

The main motivation for this work was to design and implement a testbed combining information obtained from Internet of Things (IoT) sensor data with the drivers actions within moving vehicles, with the possibility to exchange information such as video-images, vehicle's position and mobility, and other contextual data. In addition, it should set the grounds to the possibility of improving the system with information inferred from the gathered data. Therefore, the goal of this work was to provide a platform for development, implementation, and testing of novel car-assisted task utilities, network services and new sampling and transmission algorithms [10]. To achieve it, this proposal is focused in developing three specific solutions:

1. Vehicle position dissemination system (Section 3.4)

2. Traffic signal sensing for the vehicles (Section 3.5)

3. Vehicle-to-Vehicle video transmission (Section 3.7)

Figure 3.1 presents the proposed architecture, detailing the communication technology between all the nodes involved. In this figure, only two main vehicles are depicted, but other surrounding vehicles are also considered. Vehicle B (Front-vehicle) is drawn as a truck to represent the difficulty to know the front road status for a rear-vehicle (vehicle A). For ease of presentation, the camera and the CPU elements are placed only on the front vehicle and the display screen only on the rear vehicle. However, every vehicle should be equipped with both devices.

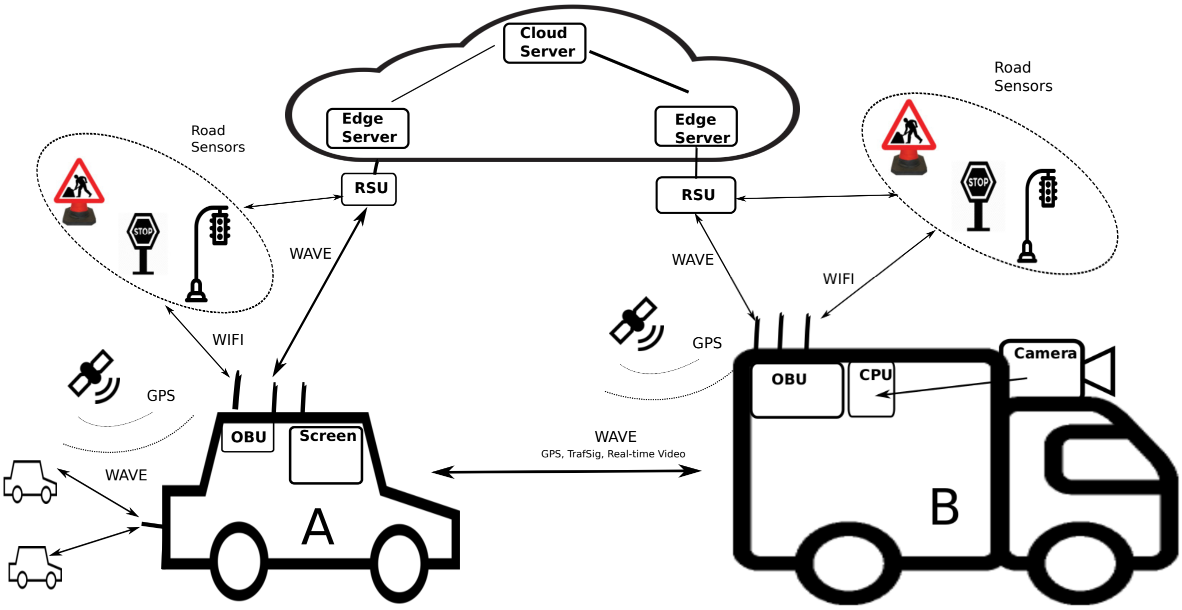

Figure 3.1: Overall system architecture.

The vehicular network used in this automotive use case consists of OBUs in the vehicles and RSUs connected to the Internet through optical fiber through an Ethernet interface. The vehicles connect among each other via standard IEEE 802.11p links, and are connected to the RSUs and the Internet through WAVE, WiFi or cellular links. In this work however, only the WAVE link is considered between the OBU and RSU. The OBUs and RSUs can establish connections with wireless devices present on the traffic signals using WiFi to collect information of their position and, if they are dynamic (such as traffic lights), their status.

Each element is described in more depth in Section 3.3.

3.3 System Elements

On Board Unit

Each vehicle is equipped with an OBU with multiple wireless interfaces, which enable the vehicles to communicate both with other vehicles circulating inside the city and with RSUs

Figure 3.2: On Board Unit.

that are integrated in the city infrastructure. An example of an OBU is depicted in figure 3.2.

The OBU includes the following components:

- Single-Board Computer (SBC)
- DSRC wireless interface (IEEE 802.11p)
- WiFi interface (IEEE 802.11a/b/g/n)
- 4G interface
- GPS receiver
- Antennas for each technology (round antenna for WiFi and rectangular antenna for IEEE 802.11p).

The SBC contains the processing unit and is responsible for coordinating the various interfaces and access technologies. Moreover, it provides an in-vehicle WiFi hotspot for the users in the vehicles and for the sensors installed in the streets and in the vehicles.

A mini-PCI 802.11p compliant wireless interface is connected via one of the mini-PCI slots. A standard 802.11g/n wireless interface is connected to one of the USB ports of the mainboard to provide communication between the OBU and other user devices. A cellular interface is available to be used whenever there is the need to exchange urgent data and no other connection type is available. The GPS receiver is integrated with the IEEE 802.11p interface of the SBC to provide multi-channel synchronization.

The OBUs runs a Linux distribution based on Buildroot. The kernel was customized to include new features such as clock synchronization, as required by IEEE 802.11p. As Linux does not provide support for the IEEE 802.11p / WAVE norm, the ath5k driver was modified to accommodate that norm within the AR5414A-B2B Atheros chipset. The driver was further extended to meet the requirements of IEEE 802.11p/WAVE [3].

Traffic Signals

One of the objectives of this work was to develop mechanisms that allow a vehicle to collect information of the position and status of nearby traffic signals. Therefore, the traffic signals need to be equipped with devices that provide communication capabilities. As such, small WiFi modules are placed on the traffic signals. The device chosen for this application was the ESP8266, depicted on Figure 3.3. It has the following characteristics:

- WiFi module with IEEE 802.b/g/n support;
- Advanced RISC Machine (ARM) processor;
- Transmission Control Protocol (TCP)/IP support;
- Works in STA, Access Point (AP) or STA+AP modes;
- Built-in low-power 32bit CPU, Serial Peripheral Interface (SPI), Universal Asynchronous Receiver-Transmitter (UART);
- Up to 16 General Purpose digital Inputs/Outputs (GPIO);
- Programmable via serial interface with Arduino IDE or NodeMCU Firmware;

Figure 3.3: ESP8266.

Display Screen

The display screen was developed for Android devices with versions from 6.0 onwards (Application Programming Interface (API) level 23). It connects to the OBU in the vehicle via WiFi and receives all the collected information of nearby vehicles and traffic signals. All this data is filtered and the most relevant is presented graphically on a map of the neighborhood. This device is also used to view the video images incoming from other vehicles whenever a request for this service is made by the driver.

Camera

The camera is used for the the V2V video streaming system and it is positioned on the front of the vehicle to record images/video from the road. We used GoPro Hero4 Black edition portrayed in Figure 3.4. It combines multiple wireless communication technologies, such as WiFi and Bluetooth, with support for video live streaming.

Figure 3.4: GoPro Hero 4 Black edition.

CPU

Different devices support different encoding/decoding methods for video images. Depending on the receiving device, the video may need to be converted between different formats. The SBC present on the OBU does not have enough computational power for this task, as such a CPU is needed to handle this process.

Road Side Unit

The OBUs and RSUs have a similar hardware, except for the antennas, which have higher gains in the RSUs. For these boards the cellular interface is also disabled, as they have an Ethernet interface connected to the fiber infrastructure.

3.4 Road Traffic Awareness

This service aims to provide an overview of the traffic density in the neighborhood by collecting and presenting position and mobility data of the surrounding vehicles. Their drivers can take advantage of the knowing the position of neighboring vehicles to deeply improve their decisions. The system conceived consists on different processes, created on the OBU, to treat and forward in an adequate way the received GPS coordinates from the neighbors, as well as the creation of a monitoring screen that receives the position packets from the OBU and present them appropriately.

To disseminate the data, each OBU broadcasts it's GPS information to all the nearby nodes as the Figure 3.5 illustrates. The WAVE technology through the IEEE 802.11p interface allows the dissemination of periodic data beacons where basic information is included. This information contains a unique vehicle identification code (ID), the geodesic position coordinates (*latitude* and *longitude*) along with the speed and heading of the vehicle measured by the GPS sensor.

Since every OBU broadcasts its position regularly, any given OBU can store this information to keep track of all the vehicles in the neighborhood. Therefore, the data received within beacons is stored as an entry of a dynamic data structure, called Node Status Information (NSI) table. A daemon service in the OBU scans the received WAVE beacons through this table and every time a new traffic awareness data is received, it is processed and forwarded through the WiFi interface to the display screen, encapsulated in User Datagram Protocol (UDP) packets. The OBU also collects and forwards to the screen the position from the car itself with the last GPS measure available. At this stage, the screen receives this data

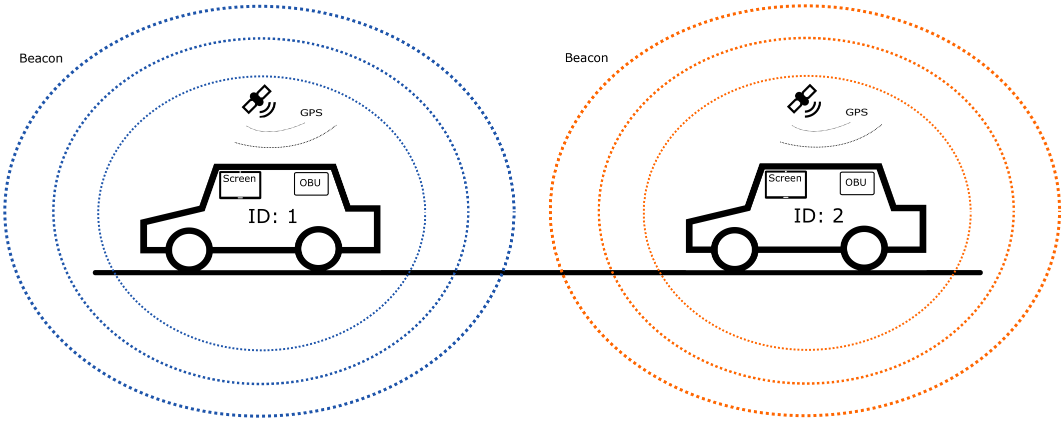

Figure 3.5: Each OBU broadcasts it's GPS information

and stores it in a dynamic data structure. After a data selection phase, the various nodes are displayed on a neighborhood map. This process is depicted in Figure 3.6.

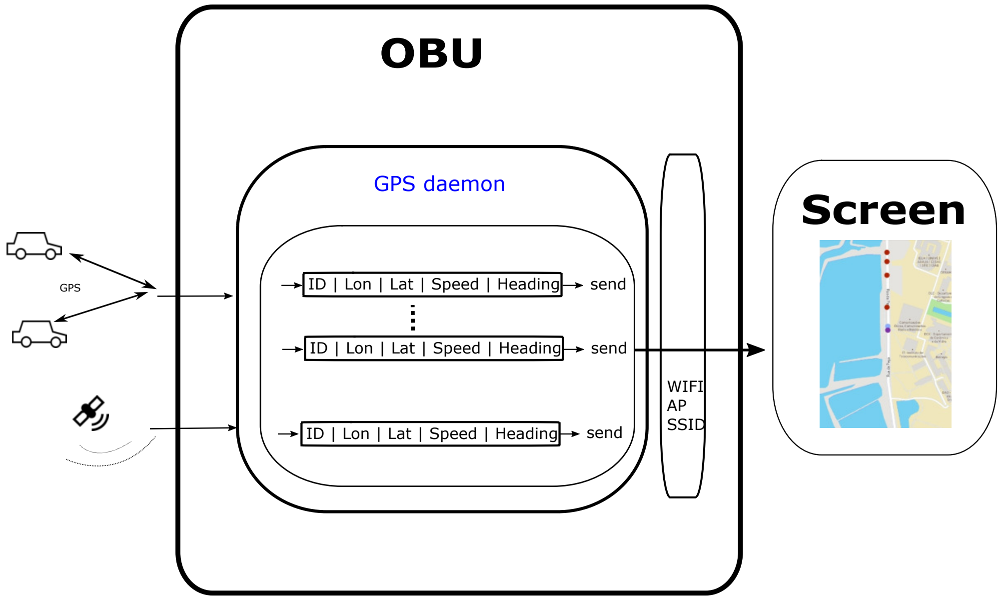

Figure 3.6: Road Traffic Awareness process

3.5 Traffic Signals Sensing

The ability for a vehicle to collect sensing data from the road status through sensors installed on the traffic signals and traffic lights, or even through data stored into the cloud and sent to the vehicle through RSUs was put into practice in the traffic signal service. This service consists on the reception by the vehicle of the traffic signal status of its neighborhood, and its dissemination to the surrounding cars. Despite being developed independently, it complements the road traffic awareness application by providing more information about the road, this time of traffic signals. The work developed involves the creation of mechanisms and communication to provide awareness of the sensors in the traffic signals. However, one main concern was to make this process adaptable to other types of sensory information and as such it also aims to set the stage for other types sensors in the streets/roads. Figure 3.7 shows the suggested architecture of the service. The hardware equipped on the vehicles is exactly the same as the road traffic awareness service, although this time the OBUs can received data from three different sources: directly from the traffic signals, from nearby vehicles or from the cloud via the RSU.

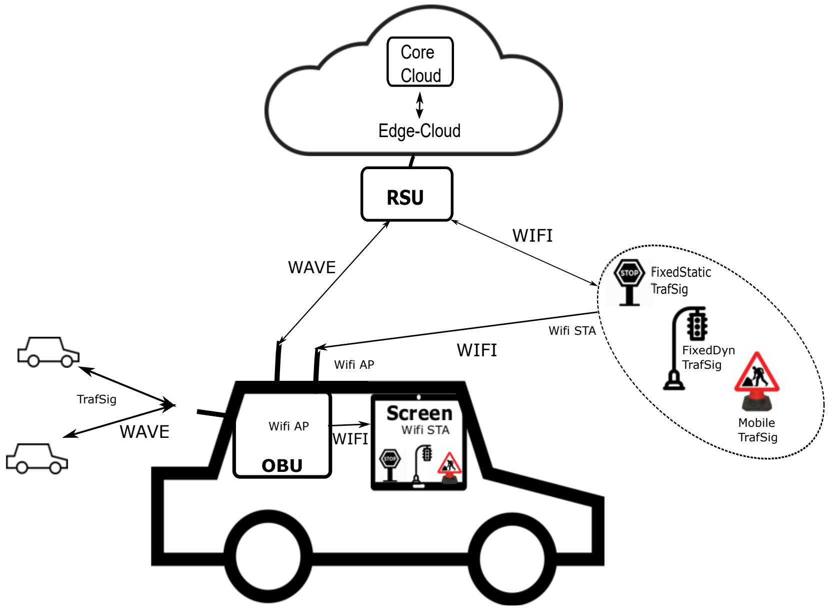

Figure 3.7: Architecture for the traffic signal sensing.

The sensors send different types of information depending on the type of the traffic signal. Hence, the traffic signals are divided in three categories:

- Fixed: like a STOP sign.
- Fixed dynamic : like a traffic light.
- Mobile: like a "work in progress" traffic signal.

For the fixed traffic signals the information is completely static, that is, the position and the status of the sign is always the same, while for the other two types either the status or the position change over time. Accordingly, for the former later cases there is a Time To Live (TTL) associated with the information, after which the information is discarded.

In an optimal solution, the communication devices placed on the traffic signals could directly broadcast their respective data using the IEEE 802.11p technology but this would make a much more costly solution. An alternative would be to broadcast the information on the ESP8266 via WiFi on an ad-hoc configuration, effectively reducing the connection time between the ESP8266 and the OBUs, however this would cause other problems. Despite being outside of the scope of this work, the WiFi interface on the OBU has to be always set on AP mode by design, as it aims to provides internet access to the passengers in the vehicle, and thus cannot be changed. As such, the interface modes would have to be managed on a Time

Division Multiple Access (TDMA) sort of way, that would effectively reduce the bandwidth available to the users of the WiFi on the vehicle. Furthermore, broadcasting this data directly via WiFi would not be ideal as it is a scenario with nodes with very high mobility.

The solution proposed is a mixture of these two. If a vehicle is close enough, the ESP8266 placed on the traffic signal can connect directly with the OBU via WiFi and transmit the information regarding the traffic signal itself. The OBU can therefore relay this data to the display screen where it is presented graphically, along with the data collected from the nearby vehicles, on a map. The OBU directly connected to the ESP8266 also broadcasts the data received via WAVE to make this information available to all the vehicles in the area. To avoid multiple broadcast storms with duplicate data, the ESP8266 only connects to one OBU at the time and always waits enough time for the data to be disseminated via WAVE, as shown in Figure 3.8. The criterion used to establish a connection with an OBU is to always choose the one with the better Received Signal Strength Indicator (RSSI).

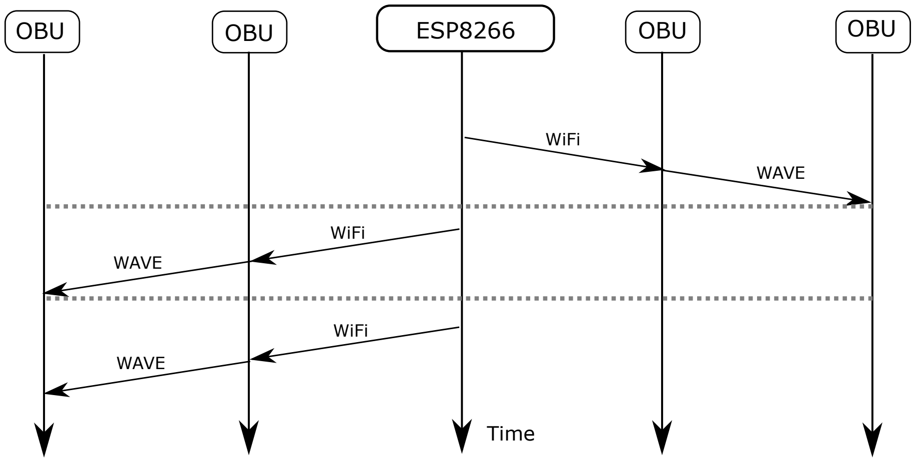

Figure 3.8: Sensory data flow.

Since the data associated with the traffic signals is not very variable and its only relevant for vehicles that are relatively close (a couple of hundred meters at most, which is covered by the IEEE 802.11p range), the dissemination method conceived is therefore rather straightforward, as it essentially is a basic broadcast with some additional control setup to prevent multiple broadcasts of the same data, at the same time.

Another way of collecting the traffic signals data is via the RSU. The RSUs can be used to provide the location of fixed traffic signals at any given point as it is a static information. However it is still assumed that not every node has this data, therefore a fixed traffic signal's data is still disseminated via the ESP8266 and OBUs.

3.6 User Feedback

To provide feedback to a driver, the data collected from the neighboring vehicles and traffic signals is funneled to a display screen where it is represented on a map of the street. The application designed for this device consists of several mechanisms to receive, select and present the data collected by the two services proposed in the previous sections. The following sections will present an overview of such processes.

3.6.1 Connection Setup and Control

It is important that the data collected reaches the screen as fast as possible to avoid unnecessary delays in its representation. Therefore, all the data collected in the OBU is sent to the screen using UDP packets. Since UDP does not guarantee packet delivery or provide acknowledges, a simple control protocol was conceived to prevent long connection loses. The Figure 3.9 portraits this process.

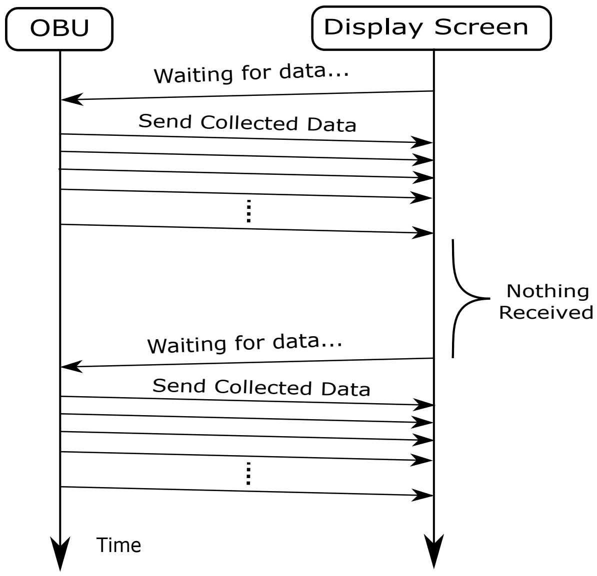

Figure 3.9: Packet flow of the OBU and display screen.

Initially the screen device connects to the WiFi network of the OBU just like any other regular connection. At this point, the OBU does not necessarily know if the connected device will actually want the collected data, as it might just be any device of the passengers connecting to get Internet access. Therefore, the OBU will only send the collected data after a request packet has been received. When this packet is received, the IP address of the sender is registered and the data packets are sent non-stop to this address. The use of packet acknowledgements seems unnecessary, as the vast majority of the packets are well delivered.

However, if a connection break happens for whatever reason, there is a significant chance that the display screen will never receive the data packets. This happens because when the display screen reconnects back to the OBU it is very likely to get a different IP address. For this specific case, the application is able to detect that an abnormally high amount of time has passed without any packets received and will periodically resend the request packet to the OBU to restart the data reception process.

3.6.2 Data Selection

Figure 3.10 shows the suggested presentation for the vehicles position. In the center of the map, the purple node corresponds to the user, while the gray nodes are the neighboring vehicles. The knowledge of the user's position is mandatory for the data representation to work as the "camera", that is, the perspective in which we see the map, is always following the user's movement, centered around the position of the user's vehicle. While this representation displays the position of the neighboring vehicles, some simple data selection and labeling mechanisms were conceived to provide some additional information associated with these nodes.

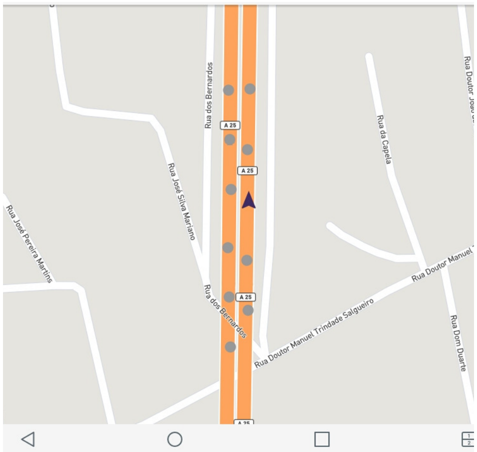

Figure 3.10: Suggested Presentation.

Vehicle Movement Detection

When viewing the position of a lot of vehicles at the same time on a map, it can become confusing to understand which is the direction of their movement. Since every OBU has access to the GPS positions of all the nearby vehicles, some calculations can be made to estimate the direction that they are moving.

Figure 3.11 represents two vehicles moving in different directions. If the heading of vehicle B (β_B) received in vehicle A (with heading β_A) satisfies equation 3.1, then the vehicle B is traveling in the opposite direction of vehicle A. That is, if the opposite of the course of vehicle B ($\beta_B - 180$) is in the green heading zone of vehicle A.

$$|\beta_A - ((\beta_B + 180) mod 360)| < \tau \tag{3.1}$$

Similarly, the movement of vehicles moving on the same lane can be identified following the equation 3.2.

$$|\beta_A - (\beta_B)| \bmod 360 < \tau \tag{3.2}$$

The usage of these methods on the previous example can be seen on Figure 3.12. The vehicles traveling on the same lane of the user are displayed in green, while the ones moving in the opposing lane are red.

Vehicle Relative Position

A distinction of the relative position of vehicles can also be estimated, particularly if they are in front of or behind the driver's vehicle. This distinction is only applied to the vehicles whose movement direction was successfully identified. If the GPS position of the user's

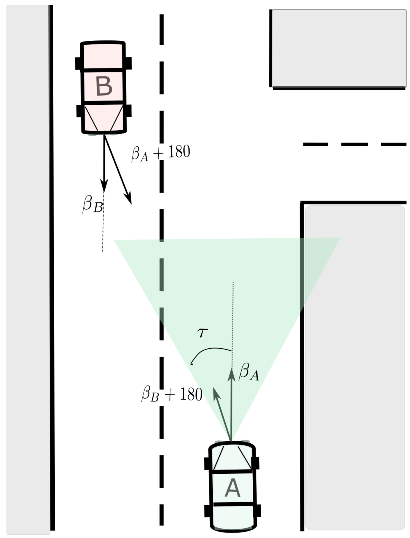

Figure 3.11: Movement Detection.

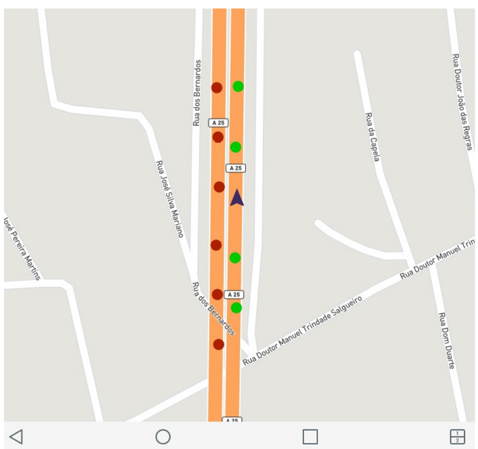

Figure 3.12: Suggested representation taking into account the vehicles movement.

vehicle is (Lat_1, Lon_1) with heading (β, and a neighboring vehicle's position is (Lat_2, Lon_2), the heading (γ) between these vehicles can be determined using the equation 3.3.

$$\gamma = (\arctan(\frac{\sin(Lon_2 - Lon_1) \times \cos(Lat_2)}{\cos(Lat_1) \times \sin(Lat_2) - \sin(Lat_1) \times \cos(Lat_2) \times \cos(Lon_2 - Lon_1)}))mod360 \tag{3.3}$$

Essentially, the heading γ yields the direction that the neighboring vehicle is facing, in the perspective of the user's vehicle. As such, the neighboring vehicle is easily determined to be in the front of driver if the equation 3.4 is met. Similarly, to determine if it is behind, the equation 3.5 is used.

$$|\gamma - \beta| \bmod 360 < \tau \tag{3.4}$$

$$|\gamma - ((\beta + 180)mod360)| < \tau \tag{3.5}$$

If this filter is applied to the previous example, the resulting presentation is depicted in Figure 3.13.

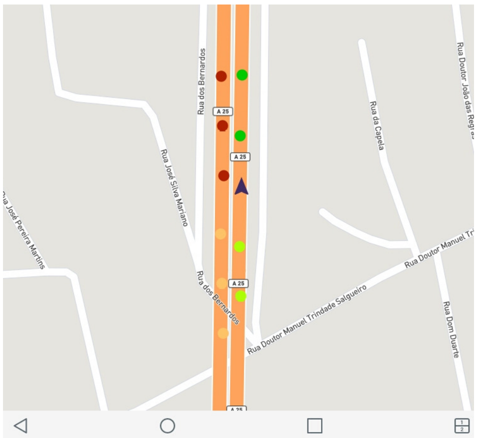

Figure 3.13: Suggested representation with both suggested filters.

Traffic Signal Representation

While the neighboring vehicles are always displayed on the map, the traffic signals are not. Generally speaking, the knowledge of the traffic signals is only relevant if they are facing towards us. As such, a selection mechanism is proposed that accounts for the direction and also the position of the traffic signals.

As seen in Figure 3.14, if the heading of the traffic light signal (α_1) is inside the green heading zone of the vehicle A, the traffic signal message will show up in the screen. This corresponds to the condition in equation 3.6. For the vehicle B, the data from this traffic signal is not considered, as it is not mapped on the yellow heading zone of this vehicle.

$$|\alpha - \beta_X| < \tau \tag{3.6}$$

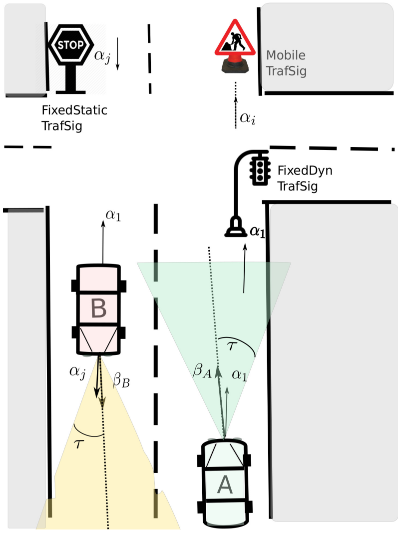

Figure 3.14: Traffic signal heading visualization.

As can be seen on Figure 3.14, the heading of the stop sign coincides with the heading zone of the vehicle B, while this information clearly is not relevant for this vehicle's driver. As such, an extra information must be taken into account, the position of the traffic signal itself. All the traffic signals that meet the condition in equation 3.6 are further tested using the same method previously described to determine the relative position between two nodes. The heading between the vehicle and the signal (γ) is calculated using the equation 3.3. If the condition 3.4 is met, then the signal is displayed on the map.

3.7 Vehicle-to-Vehicle Video Transmission

In order to expand the visibility area of a driver, a video streaming procedure allowing a vehicle to view the road as seen by the front vehicle was also conceived. This process is completely orthogonal to the previous two presented, as the data types are different, however it aims to complement the information already involved. The work developed includes the creation of communication mechanisms to provide on demand video transmission, which includes a coding process to be executed in the OBUs.

Figure 3.15 portraits the main use-case of this service. Whenever there is a large vehicle on the road, it takes out the road visibility from the vehicles behind, making it difficult for a driver to properly assess whether an overtaking action is safe. The proposed system allows a driver to request images (either photos or video) incoming from the large vehicle, to check if it is safe to overtake.

The video assistance is provided by a camera located at the front vehicle along with a CPU that processes all the necessary codifications. This data is then relayed to the OBU which will transmit it to the vehicle that requested the video images, where they can be observed on a presentation screen. Figure 3.16 illustrates the main functions of each entity involved, when transmitting the video. The images are provided by a camera located in the front of every vehicle. Should there be a need to convert those images or video to different formats (different encoders or containers), there is a CPU that will handle the video encoding process.

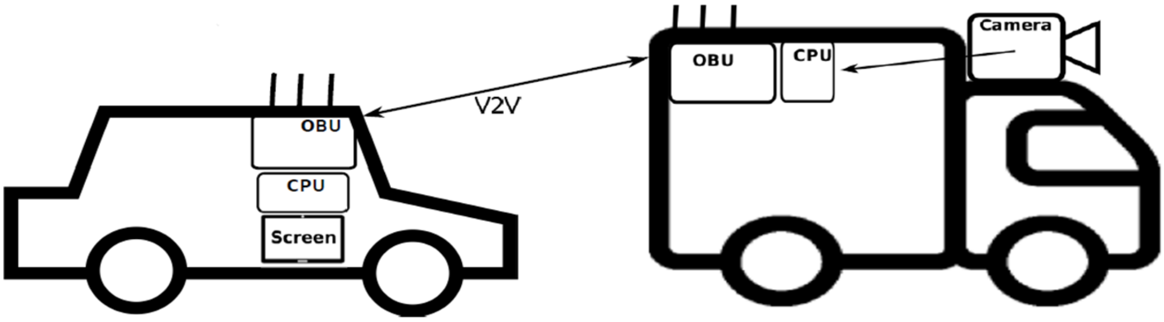

Figure 3.15: Typical use-case.

A streaming server is also deployed, which serves as a Hypertext Transfer Protocol (HTTP) server. This offers multi-platform support, as only a browser is required to watch the video content in the receiving car. Even though the overhead introduced by transmitting this data over HTTP is considerable, it is a service that is not supposed to be running on every vehicle at the same time, but rather at very specific situations like the use-case presented.

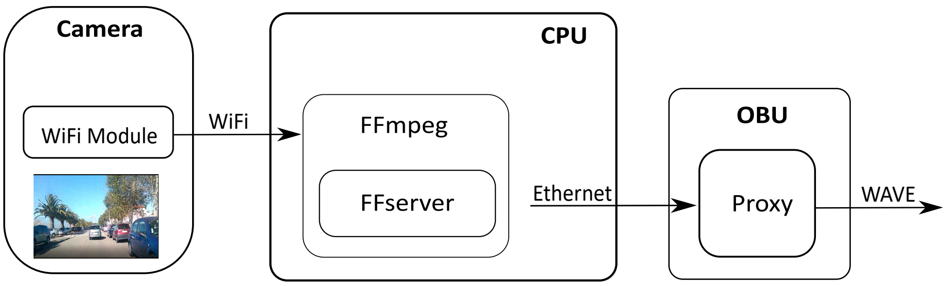

Figure 3.16: Software architecture of the video transmission.

Whenever there is a request from another vehicle, the image data is relayed to the OBU, which will transmit it to the vehicle that requested the video images, essentially acting as a proxy for the data transmission. The process for the reception is depicted in Figure 3.17. The data is sent via the IEEE 802.11p interface of the OBU which again acts as a proxy. The data is then relayed to the display device, displaying the images on a browser.

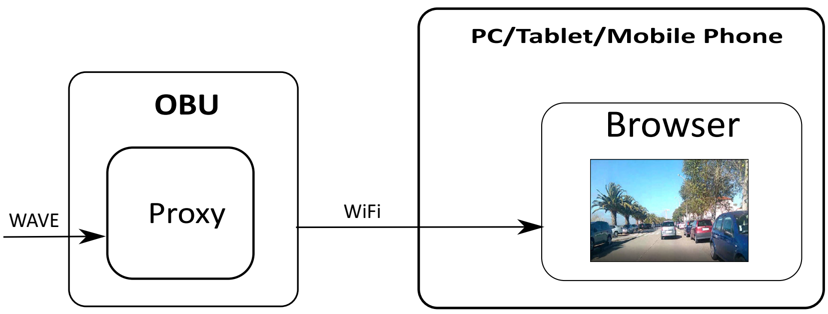

Figure 3.17: Software architecture of the video reception.

An overview of the video codification and transmission process is portrayed in Figure 3.18.

The process starts with the request of the video images, that corresponds to a HTTP get of the video data. The request is relayed via the WAVE interfaces on the two vehicle OBUs to the CPU equipped on the target vehicle. Afterwards, the CPU sends the last received video frames encoded to a different format, in the form of a HTTP post.

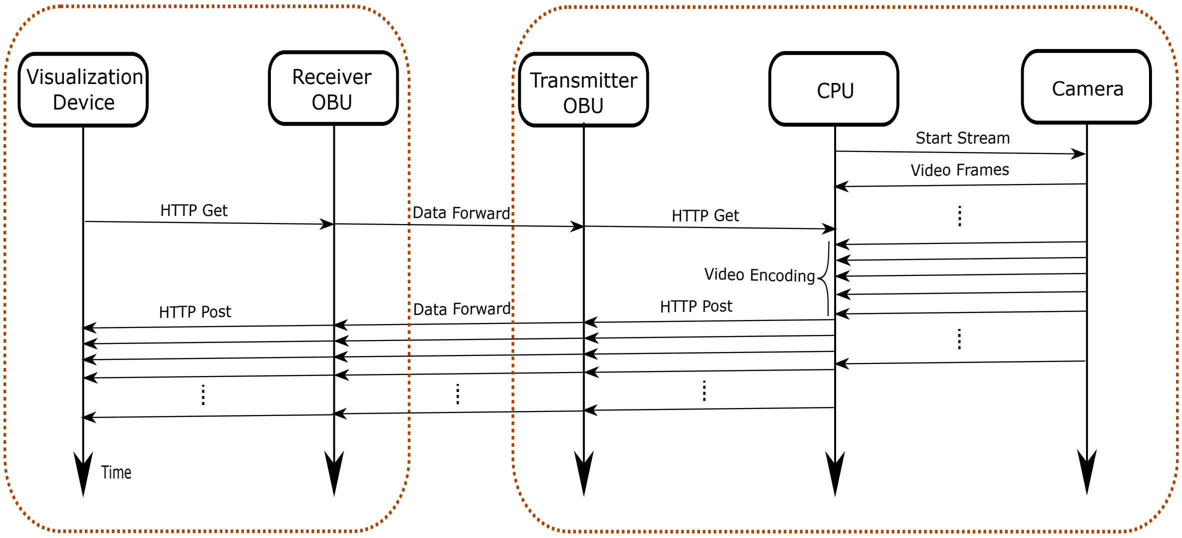

Figure 3.18: Video transmission overview.

3.8 Summary

This chapter started with the definition of the overall architecture proposed along by a brief description of the elements involved in the network was presented. After this, the three suggested solutions were discussed starting by the description of the service that collects positional data of nearby vehicles, followed by the presentation of the suggested network for the collection of traffic signals data. Afterwards, the suggest mechanisms to collect, interpret and display the data collected were presented. This chapter ends with an explanation of a system capable of recording and transmitting video images between vehicles.

Chapter 4

Integration and Implementation

4.1 Introduction

This chapter will discuss the implementation of the proposed services, mechanisms and architectures described in the chapter 3. The approach chosen for the description of the developed processes throughout the different parts of this dissertation is more action-based, with process flowcharts and state-machines.

This chapter is outlined in the following manner:

- Section 4.2: Details how all the information incoming from the multiple parts is integrated.
- 4.3: This section provides an overview of the methodologies used to collect the GPS position of the vehicles.
- 4.4: Describes the development of the dissemination mechanism proposed for the sensory information.
- 4.5: Presents the implementation details of the display screen.
- 4.6: The codification and transmission strategies developed for the video transmission are discussed in this section.

4.2 Overall Integration

This section presents an overview of the software architecture with the proposed solutions combined. The only mutual elements of all these services are the OBU and the display screen, hence these two elements needed to be adapted in order to accommodate all the involved parts. It should be noted that the V2V video transmission was developed as a parallel feature for this system, making the integration with the remaining parts rather transparent and therefore will be only briefly discussed in this section. Section 4.2.1 presents the suggested software architecture running in the OBU while Section 4.2.2 discusses the software conceived for the display screen.

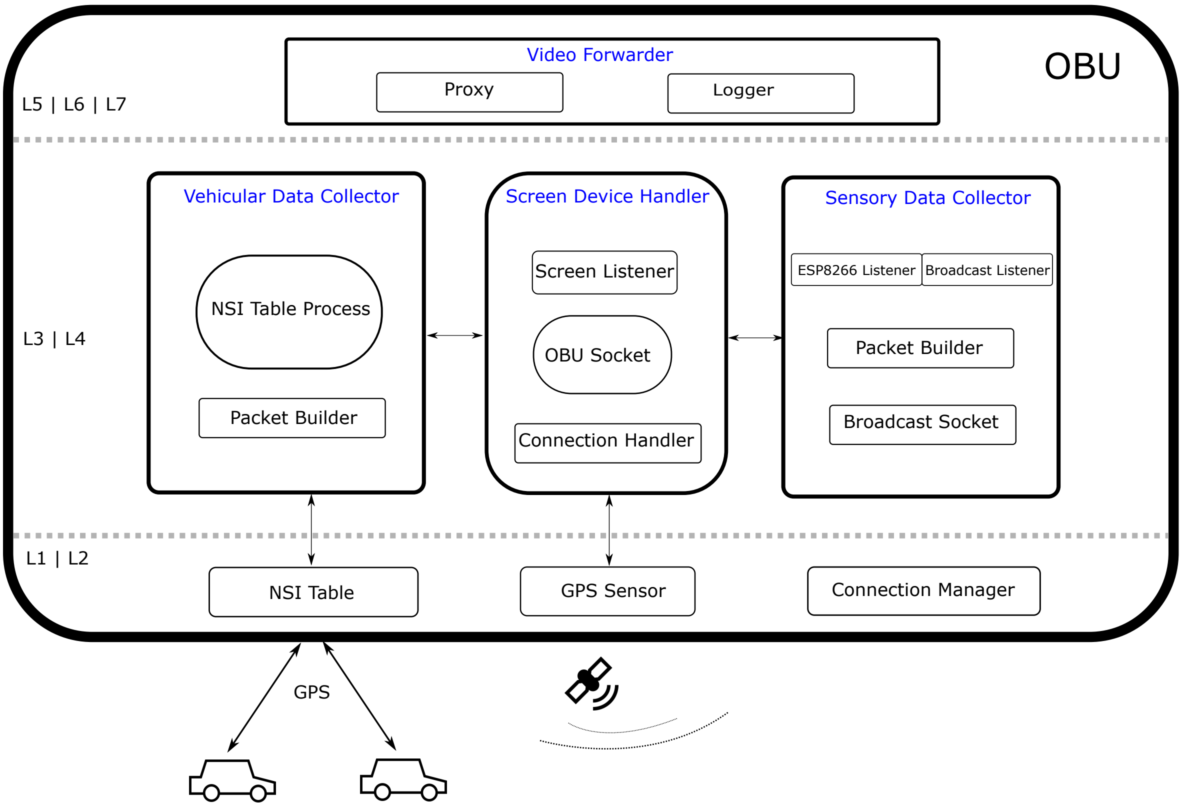

Figure 4.1: On Board Unit software architecture.

4.2.1 On Board Unit

Figure 4.1 shows the software architecture of the OBU with all the different modules developed. The data collected by the Traffic Awareness and Traffic Signal Sensing complement each other and thus these modules are integrated to essentially function as a whole. A separated module is therefore created to receive the data from these two parts and handle the communication with the display screen.

A brief description of each module is now presented:

- **Video Forwarder**: Module responsible for forwarding the video data to the service requester. As mentioned previously, the functional behavior of this module is independent of the rest as it is effectively running on a higher layer. It is comprised of a proxy algorithm that relays the video images to either the driver screen or to the OBU that made the video solicitation, and a logger that is continually saving information about the data transmitted.

- **Vehicular Data Collector**: Interprets and compacts all the vehicular data of the neighboring vehicles available on the NSI table. A packet is generated for each entry of the table, that is, for each vehicle nearby, and forwarded to the Screen Device Handler module.

- **Sensory Data Collector**: Contains two UDP socket listeners as it is responsible for the reception of data regarding the traffic signals either from the ESP8266 or from the

WAVE broadcast incoming from another OBU. After the reception of such data, it is sent to the Screen Device Handler. If the data is received directly from the ESP8266, then this data is also sent in broadcast via the WAVE interface encapsulated by UDP.

- **Screen Device Handler**: This module handles the connection and the data exchange between the OBU and the screen device. It receives the information collected by the previously two mentioned modules along with the latest GPS position received via the GPS sensor equipped on the board, and forwards it encapsulated by UDP to the screen. As all the data exchanged is via UDP, there is no direct control if such data is well delivered. Therefore a small control module was created to provide additional features such as occasional acknowledges and a reconnection ability.
- **Node Status Information table**: The NSI table keeps a list updated with all the positional information sent by surrounding vehicles.
- **GPS sensor** A GPS receiver is used to continually record the vehicle's position.
- **Connection Manager**: The connection manager falls outside of the scope of this work, nevertheless it is a major part in the functional behavior and control of all the network interfaces available in the OBU and thus is also necessary.

4.2.2 Display Screen

The software architecture of the display screen is represented in Figure 4.2. Several modules were developed to allow the reception and representation of all the collected data.

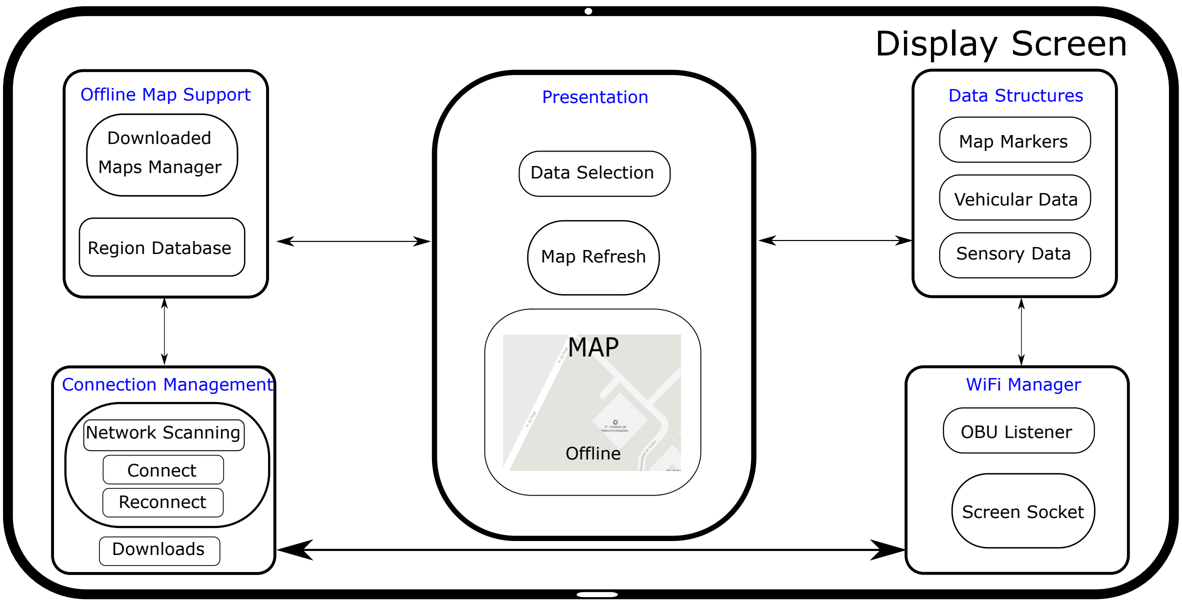

Figure 4.2: Display screen software architecture.

The main features of each modules are now presented:

- **Connection Management**: This module handles all the connections on the device. It is used to download the offline maps and to establish and control the connection between this device and the OBU.

- **Offline Map Support**: The maps deployed in this work can either be online or offline. An online map is continually downloading the tiles of the map, depending on the user's position. The concept of an offline map consists on the anticipated download and storage of a certain region. This data can be used for the representation of the map but only that region is available. Essentially, the use of offline maps is preferred as it saves a lot of bandwidth as any given user is bound to travel a lot within the same territories. Nevertheless, online maps are also available, provided this device is connected to the Internet via the OBU. Overall, this module provides the support for a user to select and download a map region for future use, forwards this data to the Presentation module whenever it is needed and also handles the storage of all the regions downloaded.

- **WiFi Manager**: The WiFi communication with the OBU is provided by this module. To receive the data, an UDP socket is continually listening for packets incoming from the OBU. The information contained in these packets is forwarded to the Data Structures. A UDP socket is also used to send control packets provided by the Connection Management module to directly communicate with the aforementioned Connection Handler in the previous section.

- **Data Structures**: All the data structures used are managed by this module. Three dynamic hash-tables are deployed, containing the information regarding the position of surrounding vehicles, sensory data from the traffic signals and also the map markers used by the presentation module.

- **Presentation**: This module is responsible for the representation of all the collected data from the Traffic Awareness and Traffic Signal Sensing solutions proposed. The data is represented in a Mapbox map of the neighborhood. The map implemented has a priority set to request and use the tiles of the offline map data first, online data will only be used if such region is not on the database. In a scenario where the vehicle is going from a downloaded region to one that is not, the transition is unnoticeable as the map gathers all the nearby tiles data, not just the tiles being presented.

  The vehicular and sensory data is extracted from the correspondent data structure. This data goes through a series of simple evaluation processes that determine if the data should be displayed and also how it should be labeled. Each entry of these lists deemed relevant enough to be displayed is then attached to an entry on the marker data structure. All of the markers' data is continually updated with the latest information available in the data structures in an asynchronous fashion. However, the refresh of the data displayed on the map itself is periodic, to provide more smooth and less confusing transitions.

4.3 GPS Position Collection

As introduced in Section 3.4, each OBU broadcasts it's GPS information to all the surrounding vehicular nodes. This process is now discussed in more detail, with an explanation of the data packets used and a detailed description of how the data collected is treated on the OBU and sent to the display screen.

4.3.1 Beacon Format

The OBUs are programmed to send periodic layer 2 data packets as part of the connection manager implemented in these boards. Even though this falls outside the scope of this work, it is important to describe this process, even if only lightly, as they carry the GPS information used for the Traffic Awareness service.

Within the context of the connection manager, they are called NSI frames and essentially act as beacons, sending important control and service advertising data. These beacons contain service information such as the Provider Service ID (PSID) and routing related data such as next-hop and endpoint ID. Among this data, the GPS data is sent as an option used for the connection manager, provided the OBU is receiving a GPS signal. Each NSI frame is sent as the payload of a IEEE 801.11 action frame, with the category set to Vendor Specific Action (VSA). Their format is depicted in Figure 4.3.

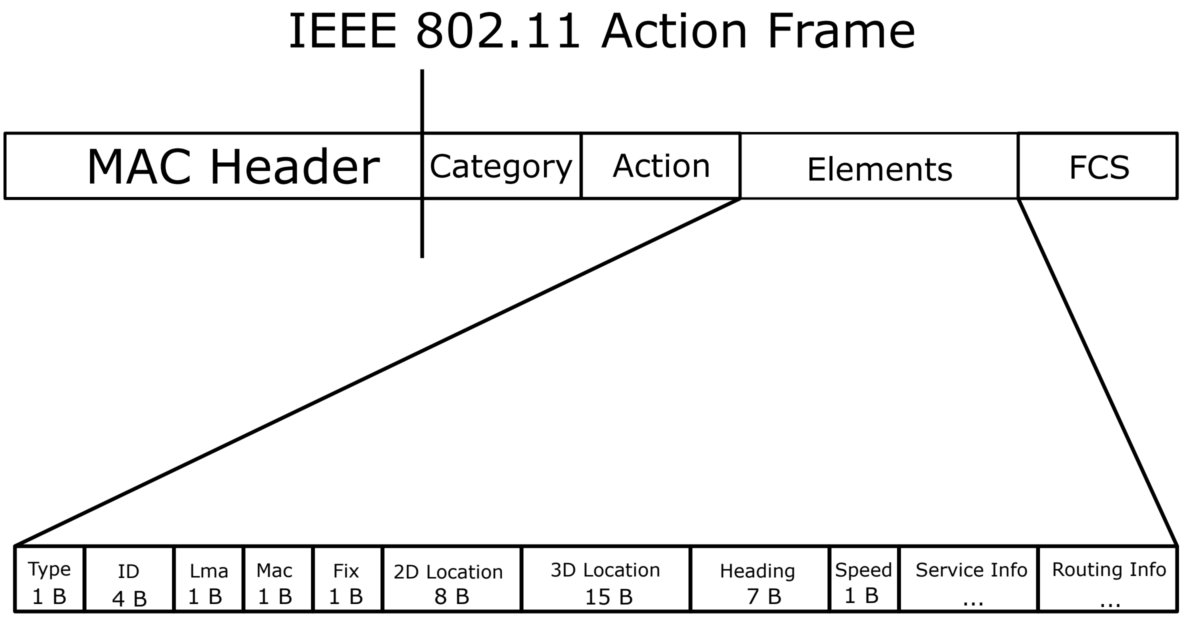

Figure 4.3: Node Status Information frame.

A description of each field is now presented:

Category Describes the action frame type. In this case it is set to VSA.

Action The action to perform. It is usually a number representing different actions.

Elements Adds extra information specific to the action.

Type Board type. Varies between OBU or RSU.

ID An unique identification code.

Lma The Lma ID. Used for mobility and multihoming control management.

MAC MAC addres of the WAVE interface.

Fix Indicates the type of GPS measure available. Either none, 2D or 3D.

2D Location Longitude and latitude.

3D Location Longitude, latitude and elevation.

Heading The heading of the movement which is given in decimal degrees, where 0^o corresponds to the geographic North.

Speed Movement speed. Indicated in meters per second.

This work adopted the use of such data packets as they are inherently needed for the overall functional behavior of the OBUs in the vehicular network environment and adding another type of dissemination method would just increase the network load without adding any real value to it. Additionally, it should be noted that these NSI frames are "closed" for the connection manager use, meaning that other types of information cannot be added (such as the information regarding the traffic signals).

4.3.2 Node Status Information Table

Upon receiving an NSI frame, the OBU retrieves all the data and stores it in the NSI table. If there is an entry already with the same ID, the correspondent information is updated, otherwise a new entry is added. An entry is considered expired if no NSI frame was received in the last 1.5 seconds and therefore is deleted after this time.

One example of the NSI table format can be seen below. For this specific case, there were two entries corresponding to two OBUs that were parked nearby.

```
List of NSI table entries:
--------------------------------------------------
Index: 0, ID: 628
Source: 30:14:4a:e9:fc:9e, Board Type: OBU
Latitude: 40.634588 degree, Longitude: -8.659453 degree
Speed: 0 Km/h, Elevation: 8.4 m, Heading: 258.0 degree
Active Service PSID: 80-6
Active Service type: Provider, RSSI: 74
Last NSI received (ms): 46
--------------------------------------------------
--------------------------------------------------
Index: 1, ID: 683
Source: 30:14:4a:e9:fc:50, Board Type: OBU
Latitude: 40.634616 degree, Longitude: -8.659453 degree
Speed: 0 Km/h, Elevation: 6.2 m, Heading: 241.7 degree
Active Service PSID: 80-6
Active Service type: Provider, RSSI: 64
Last NSI received (ms): 44
--------------------------------------------------
End of List
```

The positional information collected of the all nearby vehicles is comprised within this table. All the information carried over the NSI frame regarding the GPS information can be seen on this example, with the addition of a couple of other information fields, such as:

Index Index number of the entry.

PSID The provider service ID.

Last NSI received This indicates the time lapse between the present time (of checking the table) and the last packet received with information of this node. This number is usually comprised between 0 and 100 ms, as the default period of the NSI frames is 100 ms.

4.3.3 Position Packet Format

As mentioned previously, each entry of the NSI table is converted to a UDP packet that is sent to the screen device. An alternative implementation would be to compact all the data from the NSI table into a single packet, reducing the overhead involved for cases where there is a large amount of vehicles nearby (note that the screen device shares the same WiFi link as the passengers in the vehicle). However, since all the data is exchanged via UDP there is no guarantee of this data being correctly sent. Therefore, if all the information is grouped to one packet and that packet is not correctly sent, all the information collected in that time period would be lost. Furthermore, the size of these packets is considerably small (the payload is less than 20 Bytes) when comparing to other typical applications used by the passengers of the vehicle, meaning that the overhead introduced by repeatedly sending these packets is irrelevant.

The structure of these packets is depicted on Figure 4.4. The implementation of these packets was rather straightforward, as essentially the same GPS information received via the NSI frames is relayed to the screen. The only exception is the information regarding the elevation, which is not relevant for the data presentation and therefore is not sent.

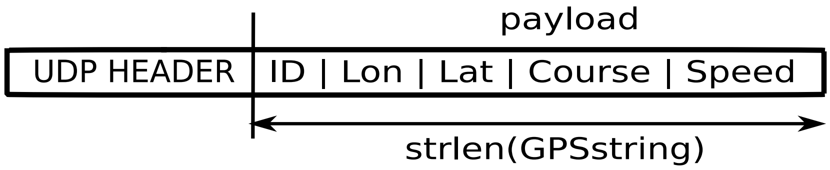

Figure 4.4: Position packet format

The same packet format is also used for the transmission of the self location to the screen. From the display screen's perspective, it is very important to distinguish the information regarding the user's position from the position of neighboring vehicles, as they are displayed differently. With the same exact packet structure used for both types of data, it would be impossible for the screen device to separate the user's data from the rest. Therefore, whenever the packet contains the data from the user, the ID field is set to a well-defined constant number (impossible to get from regular board IDs) that is recognizable by the screen as the user.

4.3.4 Process Overview

This section outlines the overall algorithm implemented in the OBUs to allow the collection of the GPS data available in the NSI table and correspondingly forward to the display screen.

Figure 4.5 depicts this process.

Initially, a configuration file is read. The configuration file is used to define a set of configurable parameters that are used throughout the execution of the algorithm. For this case, the inputs are:

T1 The time period between self location packets. Usually set from 100 to 500 ms.

T2 The time period between neighbor location packets. Typically set from 1 to 5 ms. This number should be considerably lower than the T1, to cover the cases where a very high amount of vehicles are nearby.

Once the inputs are set, the following action is to activate the inclusion of the GPS information in the NSI frames, as in some cases it is disabled. This is done by changing some parameters in the WAVE interface driver. The main loop process starts at this point. A syscall is used to obtain the lastest read from the GPS sensor on the board. If this data is unavailable, then the process restarts. As was outlined in Section 3.6.2, the user's position is mandatory for the well functional behavior of all this system and therefore is always needed. If available, then the data incoming from the GPS sensor is parsed and encapsulated into the format described in the previous section and the packet is sent. After this, another syscall is used, this time to read the NSI table. If there is any vehicle nearby then the data present in this table is parsed, one entry at the time, and the correspondent data is sent to the screen.

4.4 Sensory Data Collection

This section describes how the traffic signals sensing service was implemented, with a discussion of the packet formats used and the algorithms deployed on the ESP8266 and in the OBU.

4.4.1 Traffic Signal Packet Format

The packet format used to carry the traffic signals' data is portrait in figure 4.6. They are initially sent by the ESP8266 placed on the traffic signals to one OBU, however the same exact format is used when this OBU broadcasts the received information using the IEEE 802.11p interface.

The structure of these packets is similar to the position packets, but some of the information fields have different meanings. The different fields are defined below:

ID An unique identification code.

Lon Geodesic *Longitude* component in decimal degrees with sign.

Lat Geodesic *Latitude* component in decimal degrees with sign.

Heading The central angle where the course of the target vehicles is. The heading is given in decimal degrees where the angle 0^o is the geographic north. An explanation was given in section 3.6.2.

Message A string that informs about the nature of the signal. It always starts with "TS_" followed by the type of the signal and the status of it. For example, for a traffic light with the green light active it becomes "TS_TRAFFLIGHT_GREEN".

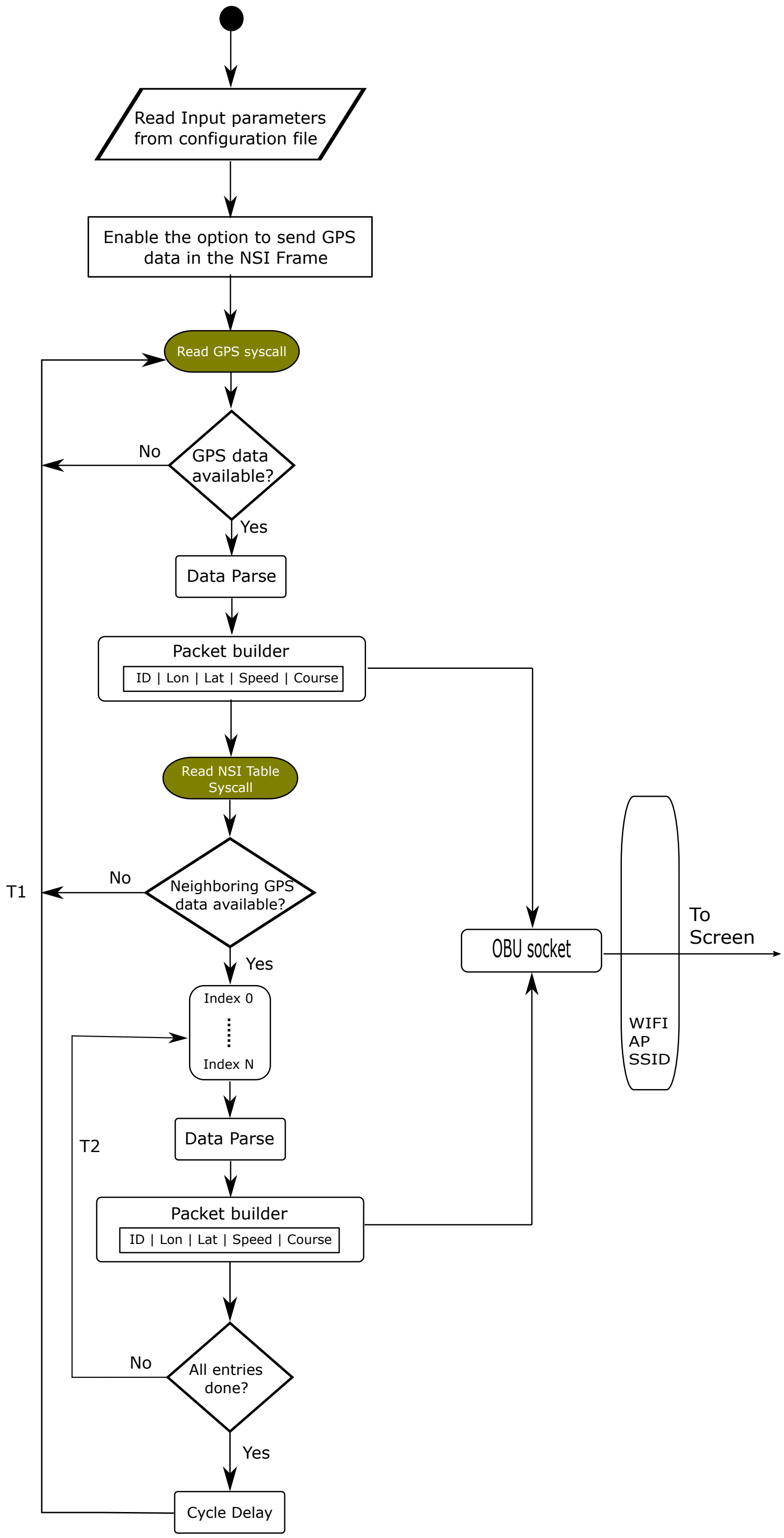

Figure 4.5: Position collection process flow

Figure 4.6: Traffic signal packet format

4.4.2 Network Scanning and Connection

As introduced in Section 3.5, the ESP8266 starts the dissemination of the traffic signals' data by sending data to the nearest OBUs via WiFi. This requires that an OBU is located close by and also not very mobile, as WiFi doesn't support mobile nodes very well. As such, the ESP will be scanning and connecting to nearby OBUs very often. As the association, authentication and address assignment procedure for WiFi links consumes a lot of time some simplifications for this process were implemented:

- The WiFi interface on the vehicle's OBU is set to always announce the network in a fixed well known radio-channel. This simplifies the scanning process by the ESP8266 as it can determine all the available networks by just scanning one channel.
- The address acquisition process by the traffic signal doesn't use Dynamic Host Control Protocol (DHCP) and it is done once the Service Set Identifier (SSID) is known.

Each OBUs reserves an unique SSID to serve as AP for passengers in the vehicle. This SSID is formed by a constant prefix "netRider" and the ID of the OBU. The ID is a four digit number that will be referenced as "1234" for explaining purposes. This number is also related with the WiFi IP network address in the form presented in table 4.1. Additionally, the devices within the vehicle that connect to the OBU and request an IP address from the DHCP server running on these boards always get a host address between 2 and 63.

Once the ESP8266 detects a valid SSID, it has all the necessary information to assign its own address as it knows the network address and the gateway, while the host address is randomly generated between 64 and 254, assuring no IP address collisions with the devices connected to the OBU within the vehicle.

Table 4.1: OBU's SSID and Addressing

SSID	netRider1234
BSS IP network	10.12.34.0/24
OBU's Wifi IP address	10.12.34.1/24
IP network Gateway	10.12.34.1

Some remarks of this process:

- The IP address is assigned dynamically based on the SSID of the OBU connected. The assignation process consist on the extracting the network address (/24) from the SSID in the way described above and the host part is randomly generated.
- There is a chance that multiple ESP8266 connect to the same OBU and the host number ends up being the exact same. However, IP address duplicity does not affect the system behavior as the OBU never sends packets to the ESP.
- In the same scenario, once the display screen receives this data from the OBU it is able to filter the data between multiple ESPs as each packet contains a unique code representing a traffic signal.

4.4.3 Process Overview

This section describes in more detail the processes involved in the dissemination the sensory data presented on the Section 3.5, by providing a description of the algorithms developed in the ESP8266 and in the OBUs.

ESP8266

The Figure 4.7 shows the functions performed by the ESP8266. The process starts with the scanning of the networks available, in the predefined channel. An SSID is considered valid when it contains the constant prefix for the WiFi network of the OBU "netRider". Once a valid SSID is found, the network and host addresses are determined following the set of rules mentioned on the previous section. After an authentication process, this device is ready to send its data to the OBU, at a rate of 1 packet per second for as long as the RSSI of the link is good enough. If it is not, then the process starts again.

On Board Unit

Figure 4.8 portraits the process running on the OBUs responsible for the reception and dissemination of the sensory data. The OBU can receive traffic signal packets in two different ways, either via the WiFi interface directly from the ESPs or by the broadcast packets from the neighboring vehicles via the WAVE interface. As such, it needs to have two processes running simultaneously to handle both types of receptions. For the packets received through the WiFi interface:

- The OBU runs an UDP socket listener ready to receive the sensory data from the ESP.
- When a packet is received it is immediately forwarded to the display screen.
- The same packet is broadcast to all the nearby vehicles.

For traffic signal packets received through the WAVE interface:

- The OBU runs an UDP socket listener to received the data sent by neighboring vehicles.
- When a packet is received, the MAC address of the sender is checked as it can actually be the packet sent from the OBU itself, when broadcasting the data received from the ESP.

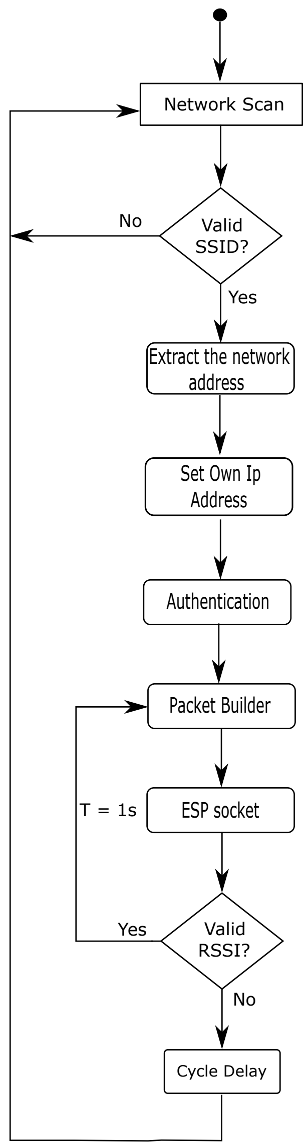

Figure 4.7: Process of the traffic signal

- If the packet is coming from a neighboring vehicle then the data is forwarded to the display screen.

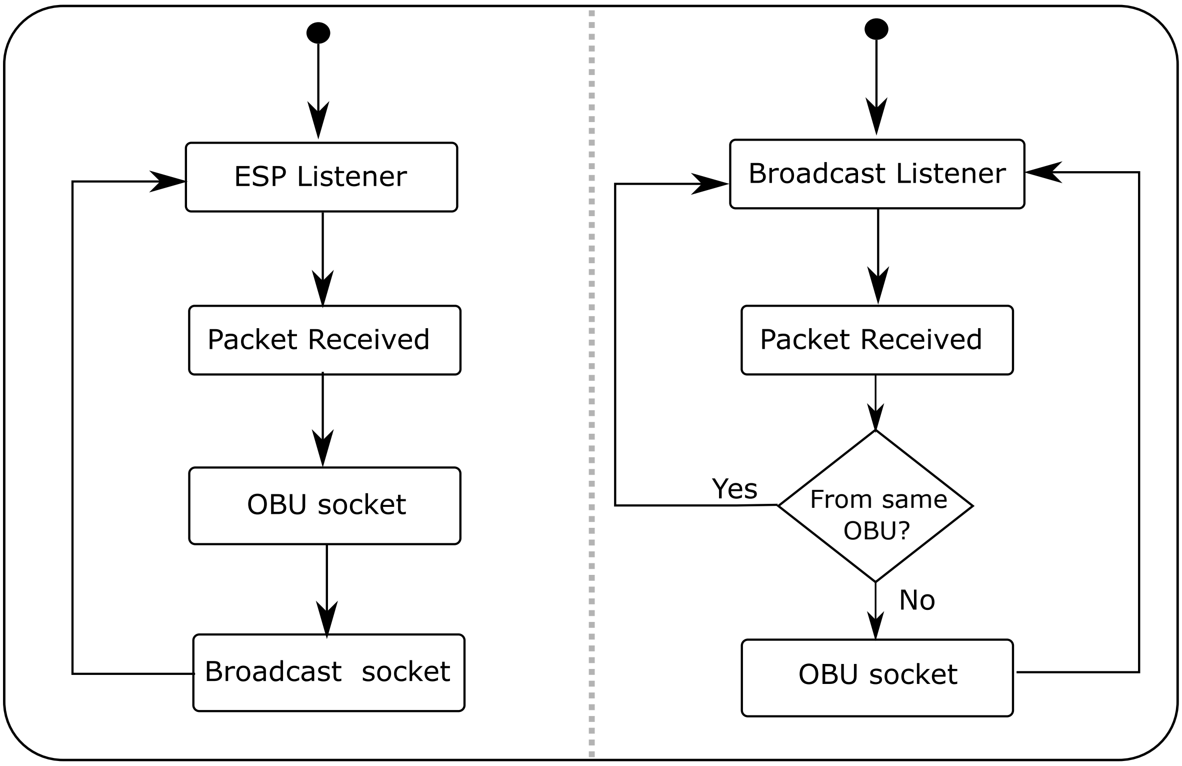

Figure 4.8: Sensory data collection flow

One objective of this work was to create a dissemination strategy that could be easily adapted to support other types of sensors on the roads. With this implementation, any sensor that possesses a WiFi module can very easily replace the ESP8266. Furthermore, even if it doesn't support wireless communications, as long as it supports UART or SPI it can be integrated with the ESP8266 to act as a WiFi module for the sensor.

4.5 Data Presentation

As presented in section 3.6, several mechanisms were conceived to create an application capable of receiving all the collected data and providing visual feedback to a driver. This section will describe the implementation of these mechanisms deployed on the display screen, with a detailed description of all the actions performed by this device.

4.5.1 Overview

The activities involved in this process are rather complex and as such will be presented in a modular fashion, starting with an overview of all the actions, as portrayed in Figure 4.9. In total, there are three different threads running on this device. The main thread is responsible for the fetching the most recent data from the data structures and for the display of such

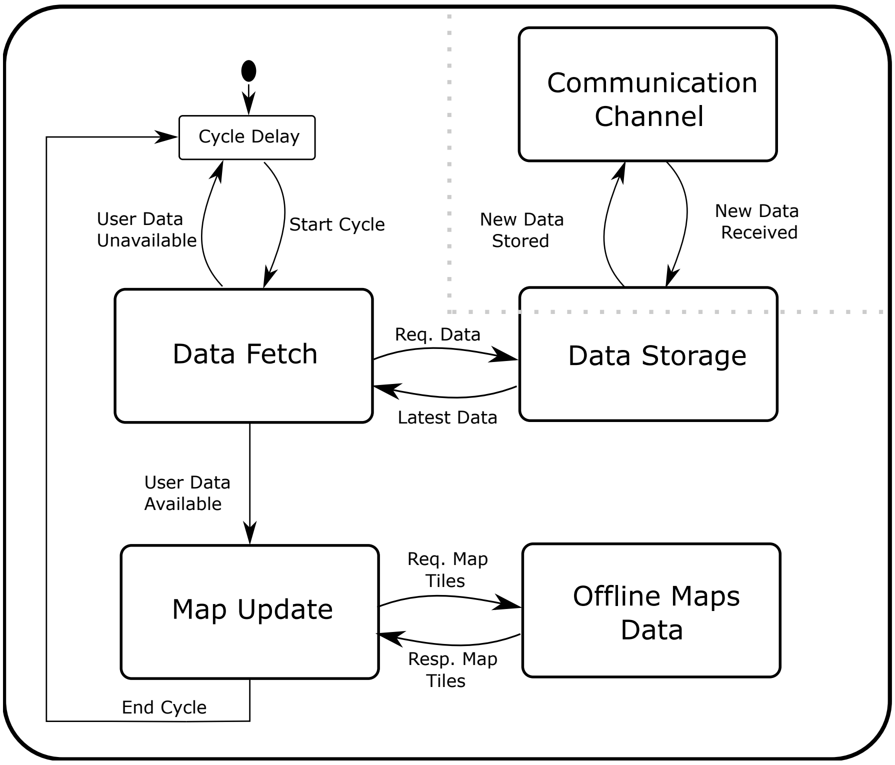

Figure 4.9: Overview of the display screen state machine.

data on the road map. A second thread is used to continually listen for the packets incoming from the OBU, while a third thread (not visible on this figure) maintains the data structures.

Overall, the main process is cyclic and it starts with the fetching of the latest available data from the data structures, which store the information received from the OBU. Once this information is gathered, it is used to update the map. To display the map itself, the use of the data stored from the offline maps is prioritized, however the option to download these maps is also available. The details of the actions involved in each one of the main states will be described in the following sections.

4.5.2 Communication Channel and Data Storage

Figure 4.10 shows the state diagram correspondent to the reception of the packets sent from the OBU. Whenever a packet is received, its contents are filtered to determine the type of the data packet, user, neighboring vehicle or sensory packet, and are subsequently stored in the correspondent data structure. Whenever an node information is updated in these structures, a TTL is attached to the entry. A separate thread keeps the data structures with updated information, by periodically decreasing the TTL attached to every entry. When this value equals zero, the correspondent entry is removed. A second action also occurs within the communication channel, to provide a simple connection control mechanism, as described in Section 3.6.1.

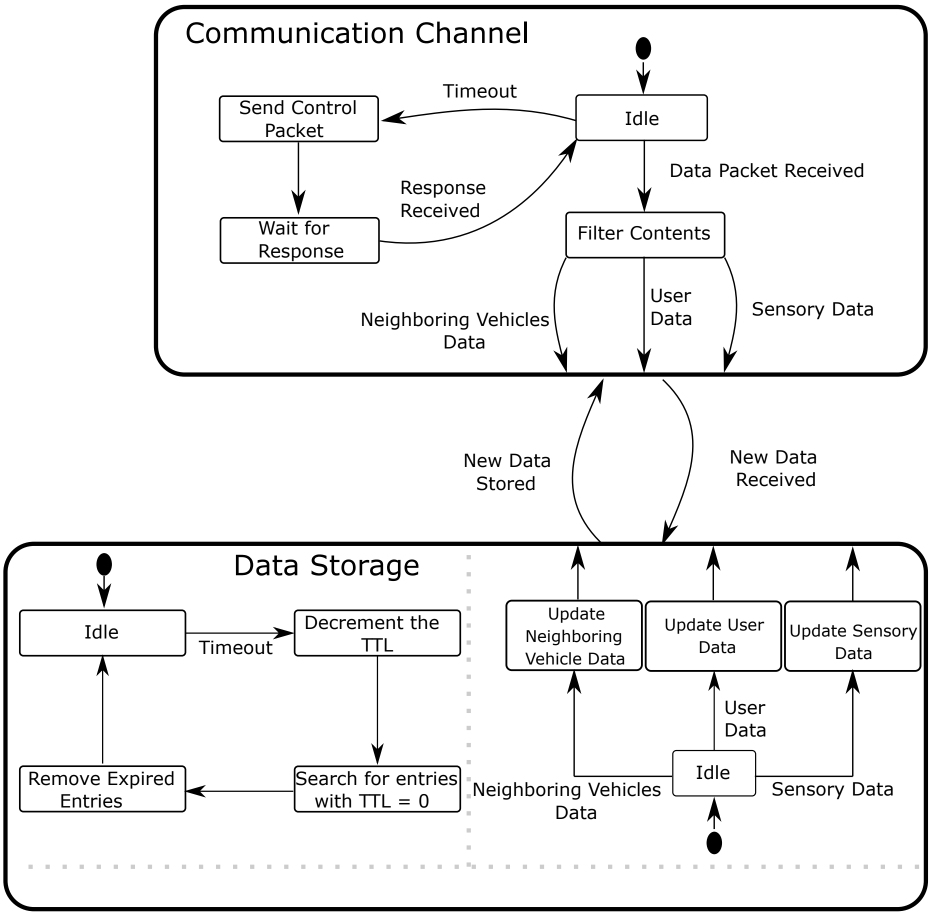

Figure 4.10: Communication Channel and Data Storage state diagrams.

4.5.3 Data Fetching and Storage

The process responsible for the fetching of the latest available data is described in Figure 4.11. The cycle running on the main thread starts here, with the request for the user data. If successful, the next step is to gather the remaining data, including the positional information from nearby vehicles, traffic signals' data and also the markers data used for the road map.

Note that the data storage state can be running on the three threads at the same time: in the main thread it is used for the data fetching from the data structures; on the second thread it is responsible for the storage of the received information; the third thread is fully dedicated to this state as it is constantly updating the data structures and removing expired entries.

4.5.4 Map Update and Offline Maps Data

The cycle ends with the update of the road map, as depicted in Figure 4.12. A series of actions are performed here: the user maker and map camera perspective are updated to the new user position. A request is sent, to get the nearby tiles of the map from the offline maps database. If available, the surrounding tiles are extracted, otherwise the tiles have to be

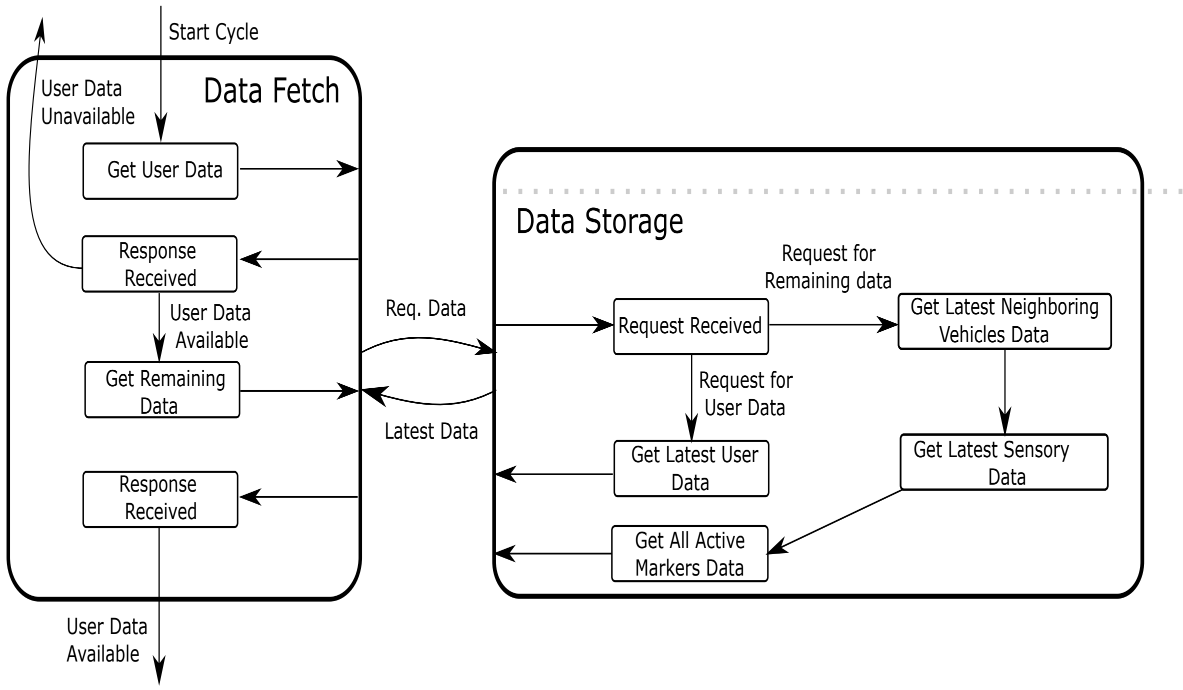

Figure 4.11: Data Fetching and Data Storage state diagrams.

downloaded using the WiFi link provided by the OBU. After this, the data selection methods discussed in Section 3.6.2 is deployed, followed by the refresh of the neighboring vehicle and traffic signal markers.

4.6 Video Codification and Transmission

This section tackles all the necessary steps in the implementation of the architecture of the video transmission system proposed in Section 3.7, from the video recording to video encoding and subsequent transmission.

4.6.1 Video Recording

The video images are provided by a GoPro camera located on the front side of a vehicle. This camera is setup to use WiFi on AP mode and establish a connection with the CPU using this technology. The camera supports live previewing on mobile devices via the GoPro App. Live previewing is the ability to see what the camera sees during a recording session in other devices, usually smartphones or tablets. Essentially, the camera has a built-in feature for streaming the video images directly to a device running the GoPro App. The GoPro App however, does not support processes such as video encoding or conversion, and thus was not a reliable option for this work. Instead, this work takes advantage of the data streaming abilities of this camera and integrates the streamed data with a video codification tool, FFmpeg.

While the GoPro stream mode was originally created to provide a preview of the images being recorded, this feature can actually be used without any data being stored in the camera's storage. Instead, the frames recorded are directly transmitted. Typically the streaming mode

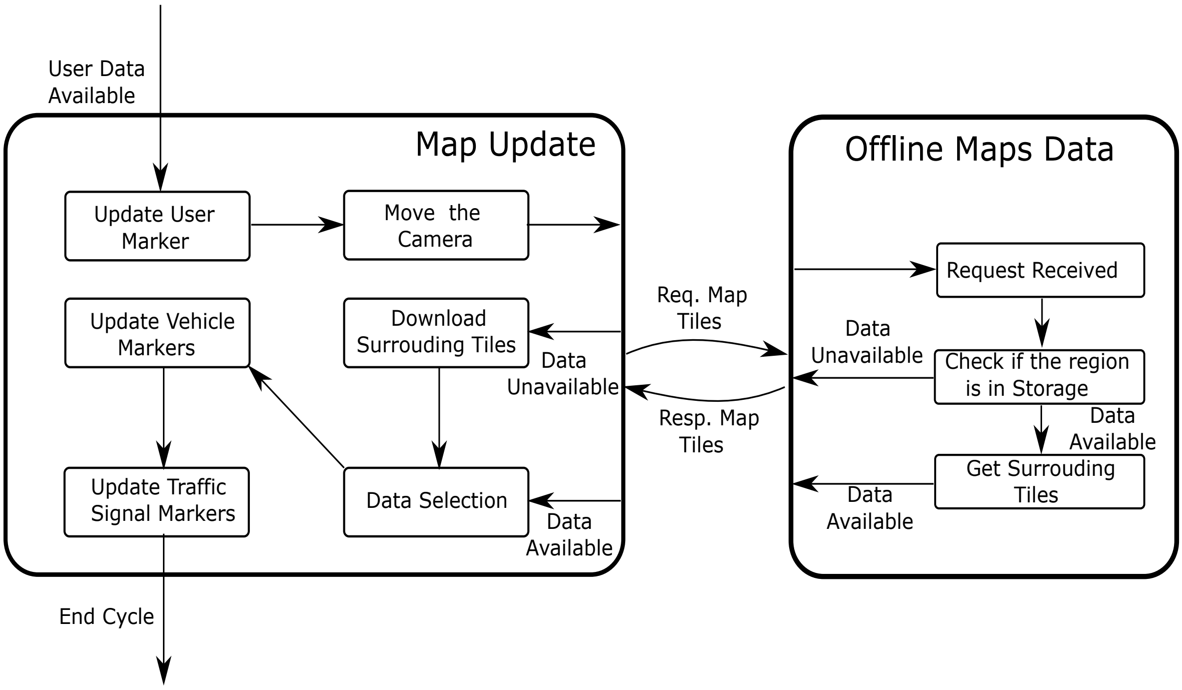

Figure 4.12: Map Update and Offline Maps Data state diagrams.

of the camera is off by default, so to start the video transmission, this mode has to be always activated first. It can be activated with the use of a control command passed by the CPU via HTTP. After this activation, the video data is then available to be transmitted using UDP on a port number specified by the manufacturer (it is port number 8554 for the camera edition used). This data will then be available for as long as the camera receives "keep alive" packets, at least once every 2.5 seconds. If this does not happen at least once, the video streaming stops and must be restarted using the previously used command to start this mode. As such, the CPU has to perform two actions to be able to receive the video images from the GoPro camera: start the stream mode using a specified control command and periodically send "keep alive" packets to maintain this mode active. Both the start up control command and the contents of the "keep alive" packets were extracted from the open source version of the GoPro App for Android and are essentially composed by a series of static control codes.

As for the video stream itself, it has the following characteristics:

Format : MPEG-TS. This is the only format supported by GoPro for video streaming. It is designed for live streaming as it divides the stream in small chunks. Additionally, the system information is sent at regular intervals, meaning a receiver can start playing the stream at any time.

Resolution : Set to 720p. A reasonable resolution considering the bandwidth availability of the scenario in question.

Frame Rate : Set to 25 Frames Per Second (FPS). A high number of frames is very important in a scenario with high mobility, as it makes up for a much more clearer and smooth image. However, a high number also means higher bandwidth usage and as such, this value was found to be a good compromise.

4.6.2 Encoding

The encoding process is deployed using the FFmpeg tool[12]. This tool is capable of converting multiple video formats, resolutions, frame rates etc. While this work adopted the used of a fixed format, it can be very easily be adapted to support other different video formats, encoders and containers. The format chosen was Motion JPEG (MJPEG), which is natively supported by most Web browsers, including Google Chrome, Mozilla and Microsoft Edge. The fact that it is supported by Web browsers means that the representation is rather straightforward, without the need of additional software. Additionally, there is no actual need to check for compatibility between the video format used by the recording camera and the presentation device. The use of this tool also allows the "tweaking" of the video characteristics, offering a way to convert the resolution and frame-rates of the video data received from the GoPro camera. Furthermore, FFmpeg is fully integrated with FFserver, a framework that essentially acts as a HTTP server, providing support to host the video data recorded by the camera. The downside of this added flexibility is that the encoding process can add a significant delay to the video transmission process depending on the computational power of the CPU in question.

As mentioned in the previous subsection, the GoPro provides the video stream via UDP in a specified port. FFmpeg was configured to use the video stream as input by specifying the format (MPEG-TS), transport protocol (UDP) and port number (8544). The decoding/encoding of the video itself is provided via a set of configurable parameters. Multiple configurations were created and tested; the following one offered the best overall behavior:

Host Port 8090.

Format MJPEG.

Resolution 720p.

Frame rate 25 FPS.

Rate control buffer size 80Mb.

Audio Disabled.

Video Intra Only Enabled.

File max size 4 MB.

It should be noted that the resolution and frame rates are purposely set to be the same as the GoPro video stream. In fact, the characteristics of the GoPro video stream were already setup having in mind that these would be the chosen parameters for the video encoding, as there would be no need for the GoPro to provide a higher resolution or higher frame rates if they would end up being wasted after the conversion process. The audio was also disabled as it is not needed for this work and it effectively reduces the bandwidth needed to transmit the video over WAVE.

The output obtained after the decoding/encoding is a feed.ffm file, that is essentially the input file for the FFserver. The FFserver takes this file and creates a new file with the format indicated and makes it available via HTTP. The new file, that is the new video, has a maximum size of 4 Mb. This is purposely set to be very low to make sure that when this video is requested, it always contains the latest data converted. If for example, the maximum file size was 1GB, then whenever someone makes a HTTP get of this video, they would see the video as when it was started to be converted, instead of the live video images.

4.6.3 Data Forwarding and Visualization

To trigger the transmission of the video between the vehicles, the receiving OBU has to send a request. The request consists on a HTTP get method of the video file stored in the CPU connected to the transmitting OBU. The Uniform Resource Locator (URL) used is composed by the IP address of the OBU, the port number of the proxy running on the OBU and the name of the video file itself. To differentiate between IP address and ports a similar process to the one presented on Section 4.4.2 was deployed. The relation between the ID of the board and the IP address is the same and a similar system is used for the port numbering of the proxy. This number is obtained by adding a constant number 10000 with the unique ID of the board, as suggested in table 4.2.

Table 4.2: IP addressing and port mapping

SSID	netRider1234
OBU's Wifi IP address	10.12.34.1/24
Proxy Port	11234
URL	10.12.34.1:11234/file.mjpg

Using this example, the proxy running on the receiving OBU would redirect the request to the proxy of the transmitting OBU with, for example, ID 5678 and consequent port 15678. This request is redirected again to the port of the HTTP server provided by FFserver, that is always fixed to be 8090. The encoded video is then able to be transmitted back to the service requester following the reverse path and the live video images will then be displayed on the browser that initiated this process.

4.7 Summary

This chapter described the main implementation details of this work. It started with an overview of the integration of these different parts coupled with a discussion of the software architecture of the main elements involved. After this, followed a description of how the GPS positions of neighboring vehicles are collected. Afterwards, the methodologies adopted for the collection of sensory data were presented. The following section described how the display screen is able to received the collected data, run a series of simple evaluation processes and then display the data. Lastly, the main implementations necessary for the vehicle-to-vehicle video transmission system was presented.

Chapter 5

Evaluation

5.1 Introduction

All the scenarios conceived and evaluations performed are described in this chapter. It is organized in the following way:

- Section 5.2 present the several tested scenarios for the data selection and consequent display of the collected vehicular and sensory data.
- Section 5.3 describes the scenarios proposed to test the video transmission between vehicles, with a detailed description of the results obtained along with a discussion of the limitations involved in the technologies and equipment used.

5.2 Data Selection and Display

There is no easy way to automatically evaluate whether the data selection methods proposed are correctly working. The only feasible way to verify whether those mechanisms are properly functioning or not, is through visual validation and as such, this section will presented in a more demonstrative manner. As methods that provide feedback to a user, a significant part of the evaluation boils down to a matter of preference. Still, some typical scenarios were conceived to test the data selection mechanisms suggested and also to calibrate all the necessary input parameters.

Movement direction detection

The mechanisms deployed for the data selection can very easily verify the correctness of the movement direction of neighboring vehicles when they are moving on a straight road, as the difference of the heading between the user's vehicle and the surrounding vehicles is very small. In this case, any given threshold τ can pretty much always guarantee a correct detection. However, this distinction is not as straightforward when considering roads with a significant curvature. The first scenario tested consists on multiple vehicles traveling in the opposite lane of a given driver, as depicted in Figure 5.1. The same color scheme is used as the one presented in section 3.6.2. The purple node corresponds to the self position. The green node is a vehicle found out to be traveling on the same direction, while the red ones are found out to be moving on the opposing direction. When the data selection process does not detect anything, the nodes are displayed grey and only their position is shown.

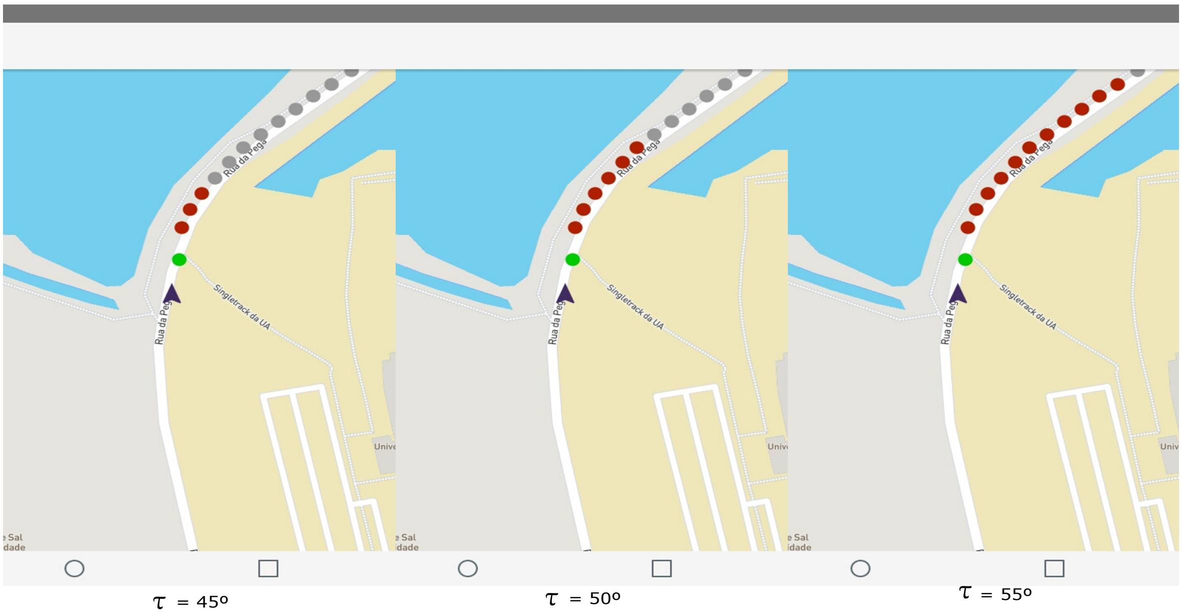

Figure 5.1: Movement display: calibration of the τ parameter.

The aim of this test was to calibrate the angle threshold τ used for the detection of vehicles moving in different directions, presented in equations 3.1 and 3.2. The Figure 5.1 shows the impact of the variation of this parameter. The first approach used a value of $\tau = 45^o$ and only a few vehicles were able to be detected as moving on the opposite direction. As this parameter rises, more vehicles are able to be detected when moving on curved roads. This cannot be continually raised though, as it would inevitably start detecting and labeling the movement of vehicles on situations like four-way intersections, leading to errors in the display process. Although in the end it is almost a matter of preference, a reasonable value found was 55^o, which corresponds to a maximum detection angle depicted on the right side of the Figure 5.1. Essentially, with the chosen angle threshold, all the vehicles moving in a road curvature as the one displayed can be detected and correctly labeled. After this angle, the data collected is displayed without any additional labeling of information.

Relative position detection

Figure 5.2 shows another typical case. In this instance, the user of this application is heading North with a surrounding vehicle in front of him. The red nodes correspond to the vehicles that were found out to be traveling on the opposing direction. Their color changes once they are identified to go past the user's vehicle. This test was used to determine an acceptable angle threshold used in the equations 3.4 and 3.5 to determine the relative position between two different nodes. As this filter is only applied to vehicles that were already categorized as traveling in the same direction or on the opposing direction of a given user, the margin for this angle can actually be higher than for the τ on the previous case. Essentially, there is only two options for these nodes, they can either be in the front or behind, there is no other real option. For this case, a τ angle of 90^o was determined to be a good fit. When looking at the equations in question, this value essentially means that every node (whose movement has been previously identified) that is within $\{-90^o, 90\}$ of vehicle's heading, is considered to be on his front; while everything that is within $\{90, 270^o\}$ is considered behind.

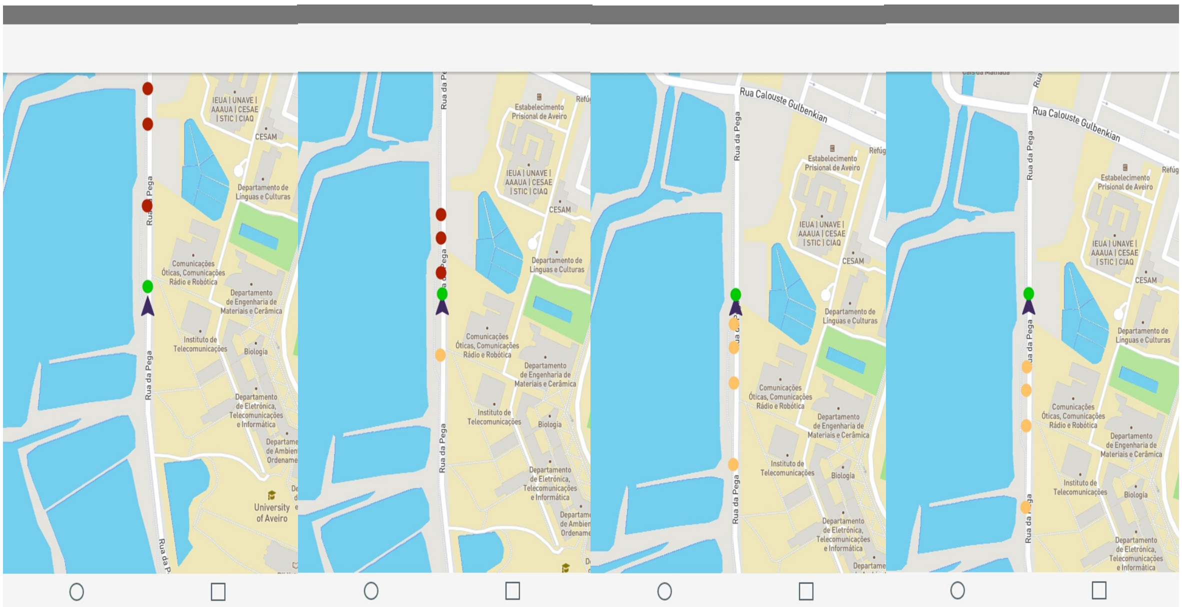

Figure 5.2: Example of the relative position display.

Traffic signaling data detection

To validate the rules presented in Section 3.6.2, a scenario was created consisting of a four way intersection with a semaphore present on each road. This scenario is portrayed in Figure 5.3 with four Points Of View (POV) of the display screens of the different vehicles. In this case, there is a vehicle positioned near each traffic light that receives the data coming from the ESP8266 installed on the signal. All the four vehicles receive the information regarding all the signals as they are within the reach of the WAVE broadcast. However, the display screen of each vehicle shows only relevant information for the driver following the proposed rules. Once again there is a considerable part of subjectivity and preference in the visualization involved, nevertheless it was determined that a match for the angle thresholds used are the same ones presented in the previous subsections. That is, to determine whether a traffic signal is in the direction of a vehicle, a threshold angle of 55^o is used and to detect whether it is in front of the vehicle an angle of 90^o is deployed.

5.3 Vehicle-to-Vehicle Video Transmission

Of all the services presented in this dissertation, the vehicle-to-vehicle video transmission system is by far the one that demands more bandwidth from the vehicular network. Additionally, the correct behavior of this system is far more critical than either the vehicular or sensory data collection, that have a reasonably higher margins for errors or delays. As such,

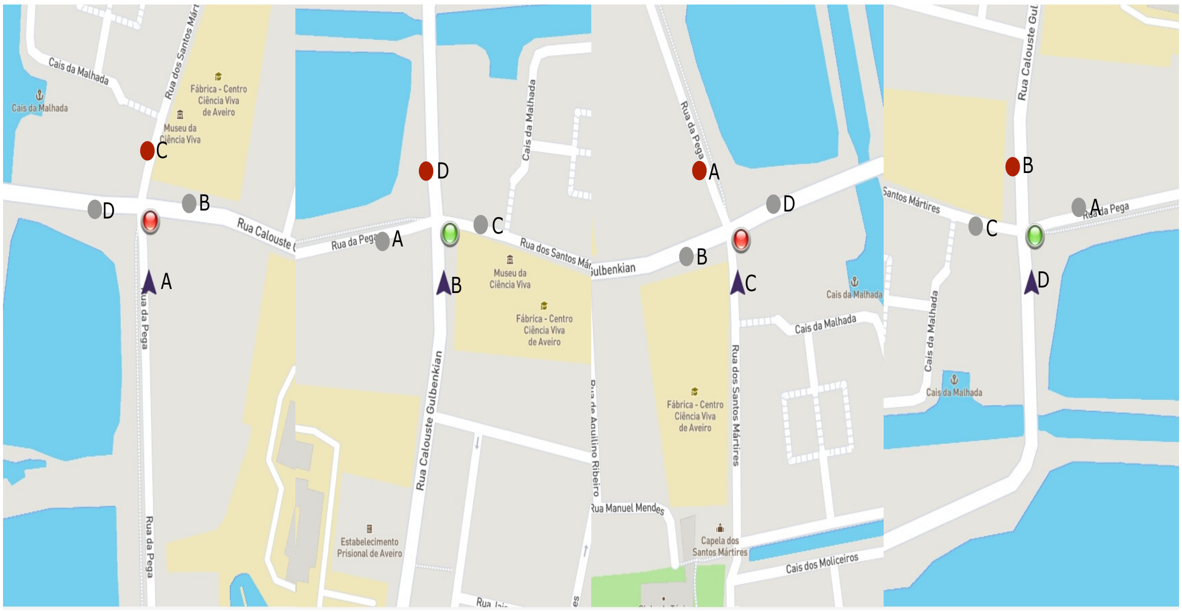

Figure 5.3: Example of the traffic light display.

---

in this dissertation the testing of this system was prioritized over the remaining parts. In order to evaluate the proposed video transmission system, two vehicles were used: a front vehicle who was continually recording the road and transmitting the video images collected, and a vehicle behind receiving the images. The video streaming system setup consisted on a GoPro camera Hero 4 black edition wirelessly connected to a Laptop with the following characteristics:

- Intel Core i5-6300HQ @ 2.30GHz.
- 8 GB @ 2133 MHz.
- Linux Mint 18.
- FFmpeg/FFserver version 2.8.11

As a reminder, the Laptop is used to transcode the original GoPro video stream format to a format compatible with Web browsers and correspondent forwarding of the new video data to the OBU. The video is then delivered over HTTP. The transcoding process is performed by FFmpeg/FFserver, which provides a simple way to convert video input while it is still being captured from a live source. These are the FFmpeg/FFserver parameters settled in this evaluation:

Host Port 8090.

Format MJPEG.

Resolution 720p.

Frame rate 25 FPS.

Rate control buffer size 80Mb.

Audio Disabled.

Video Intra Only Enabled.

File max size 4 MB.

Figure 5.4 illustrates the evaluation setup with the transmitting vehicle on the left and the receiving one on the right.

Figure 5.4: V2V video stream setup.

As it is a real-time service that can ideally be used in emergency scenarios, we have considered the communication delay as one of the most important features to be evaluated. Thus, comparisons were made between the communication delay and the speed of each vehicle, as well as with the quality of the IEEE 802.11p link.

In addition, some other important measures were taken, such as the used bandwidth and the packet retransmission rate. Furthermore, the system was evaluated under different conditions such as different traveling speeds, distances and levels of line of sight obstructions. To that end, experimental results were obtained through two different scenarios: urban and highway.

The measurements were performed in the following way. Initially, the clock of the two OBUs was synchronized using Network Time Protocol (NTP). A packet capture software Tcpdump is used in both OBUs to continually record the information of all the packets sent and received on the ports specified for this service. The packet delay and also the number of retransmissions are measured by comparing the log files obtained by these captures in the following manner: first, a packet is detected on the transmitter logs. At this moment the ID and the seq. number associated with this packet are registered and a search for these parameters is made in the receiver's records. If it is found, the time difference between the two entries is made to estimate the communication delay of the packets. It should be noted that the software used to capture the packets has an temporal accuracy of each entry in the order of microseconds, which is a reasonable accuracy considering that the order of magnitude of the transmission of these packets is around 0.5 up to 3.5 milliseconds.

To register the RSSI of the WAVE link, the information present in the table NSI presented in Section 4.3.2 was used. The signal quality measurement available in this table is a value between $0 - 100$. In this scale, when the signal strength is bellow -100dBm, it corresponds to 0; when it is higher than -50dBm it is considered to be 100. For values between -100dBm and -50dBm, the relation between the signal quality and the signal strength is the following:

$$Quality = (2 \times dBm) + 100 \tag{5.1}$$

In order to estimate the distance between the vehicles, their GPS position was continuously recorded in each OBU. With these measurements, the distance can be easily calculated. For the calculation of the bandwidth used, a program was created that accesses the driver information of the WAVE of each board and continuously records the number of bytes sent by each second. For a more correct measurement for the bandwidth used for video transmission, the bandwidth used without the use of this service was also measured. Thus, the numbers shown in this dissertation correspond to the total measured value subtracted from the value measured in normal operation without video transmission.

5.3.1 Urban Scenario

Severe obstructions of line-of-sight can result in insufficient bandwidth for video transmissions. As such, it is of crucial importance to evaluate this system in a scenario where this is a common occurrence, as is the case of an urban scenario with a very dynamic environment including a large number of roadblocks. The urban scenario is also characterized by a very high vehicular density, low traveling speeds and small distances between vehicles. For this scenario, the results presented correspond the an average of a total number of about 532000 samples, obtained during two different set of experiments. All the graphics present in this section are represented with a confidence interval of 95%.

The relation between packet delay and the speed of both vehicles is depicted in Figures 5.5 and 5.6. To better identify the influence of the velocity of each vehicle, several tests were performed with one vehicle travelling at a constant speed and the other with different speeds. For both cases, the measurements are performed when the distance between cars is similar, around 30m. More specifically, the results shown in Figure 5.5 consider that the vehicle transmitting was traveling at a constant speed of 30km/h, while the receiving vehicle moved at different velocities. The results depicted in Figure 5.6 consider the opposite case; the receiving vehicle was traveling at a fixed speed of 30km/h with the transmitting one moving at multiple speeds.

The packet delay increases with the increase of the relative velocity between the two vehicles. The delay itself is relative low as the distances between the vehicles in this scenario are usually very low. Generally, the relationship of the communication delay and the traveling speeds is quite similar for both vehicles, even though the receiving car's velocity has a higher impact for higher speeds.

The Figure 5.7 illustrates the comparison of the delay with the quality of the IEEE 802.11p link. Just as expected, this system performs better as the quality of the link increases. The minimum RSSI verified to reliably run this service is approximately 9, which was only verified on extremely crowded areas with a lot of line-of-sight obstruction.

Overall, the delay involved in the communication is relatively low, especially when considering a scenario so dynamic such as this one. Table 5.1 shows the average values for all the measurements for the V2V video transmission in the urban scenario.

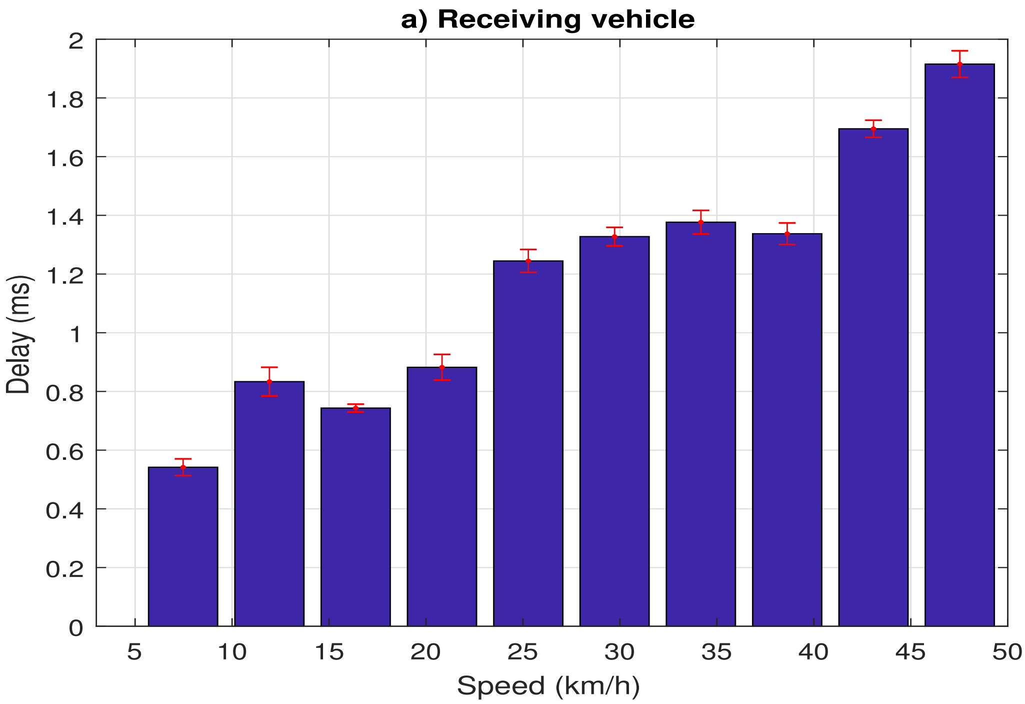

Figure 5.5: Urban scenario - Comparison between packed delay and speed of the receiving vehicle.

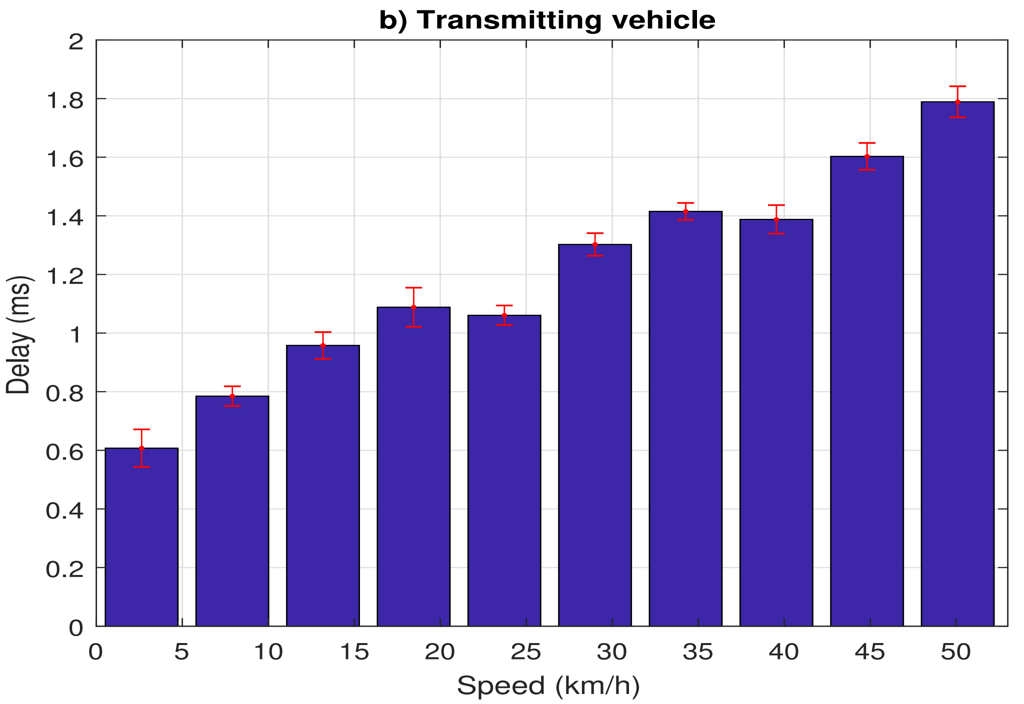

Figure 5.6: Urban scenario - Comparison between packed delay and speed of the transmitting vehicle.

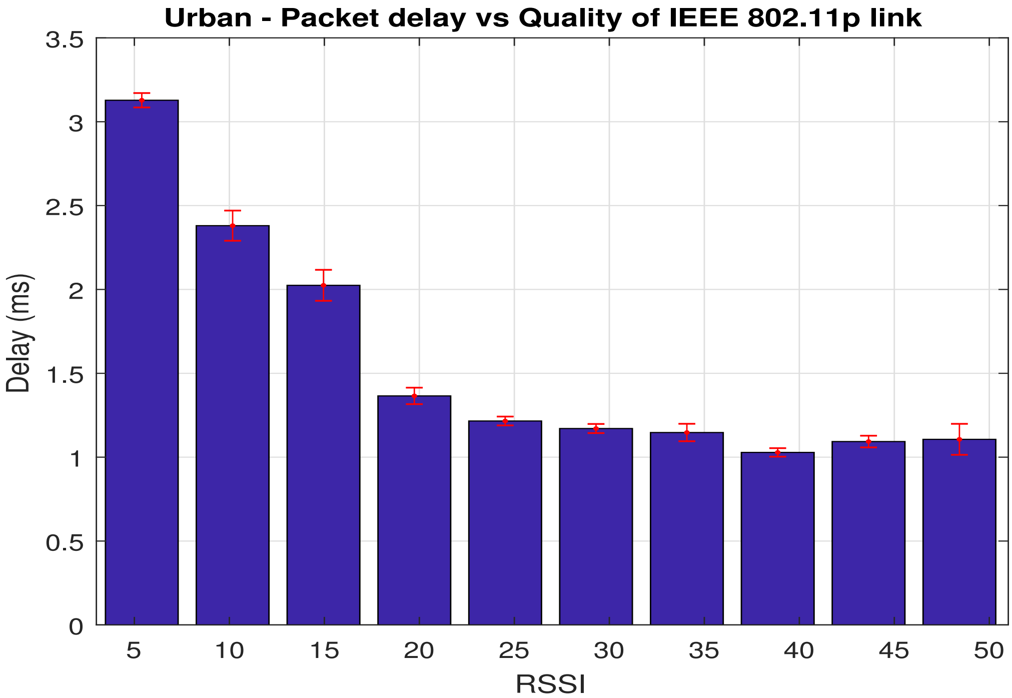

Figure 5.7: Urban scenario - Comparison between packed delay and quality of the IEEE 802.11p link.

Table 5.1: Overall average results for the urban scenario.

Distance (m)	RSSI	Delay (ms)	Bandwidth (Mbps)	Retransmissions (%)
28.45	35.41	1.22	4.71	0.80

5.3.2 Highway Scenario

This scenario is characterized by having a lower vehicle density when compared to the last one but a much higher mobility of the vehicles. The vehicles move at higher speeds with bigger distances between them which naturally leads to larger delays. For the highway scenario, the results presented correspond the an average of a total number of about 703000 samples, obtained during three sets of experiments. All the graphics present in this section are represented with a confidence interval of 95%.

Similarly to the previous scenario, to determine the influence of the speed of both vehicles on the packet delay, one vehicle moved at a constant speed while the other varies in speed. The measurements are performed when the distance between cars is identical, around 60m. Figure 5.8 represents the case where the transmitting vehicle was moving at speed of about 95 km/h. Figure 5.9 represents the opposite case, with the receiving vehicle moving the same constant speed of 95km/h.

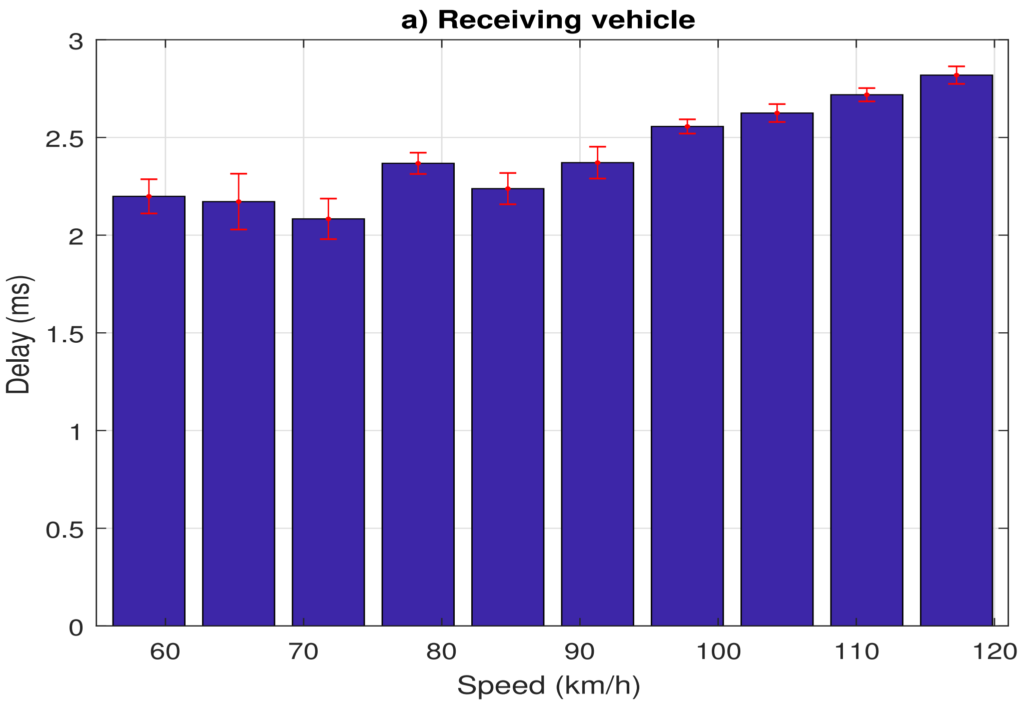

Figure 5.8: Highway scenario - Comparison between packed delay and speed of the receiving vehicle.

Just like the previous scenario, the relation is pretty much identical between the communication delay and the traveling speeds of both vehicles. Following the same trend as before, the receiving car speed has a slightly higher impact.

The relation between the RSSI and the delay is similar to the previous scenario, as shown in Figure 5.10 . Overall the RSSI has lower values because the vehicles are naturally more spaced out in a highway scenario. In this case the minimum acceptable RSSI for a stable connection verified was 13.

Table 5.2 presents the summary of the behavior of the V2V video transmission on the highway scenario.

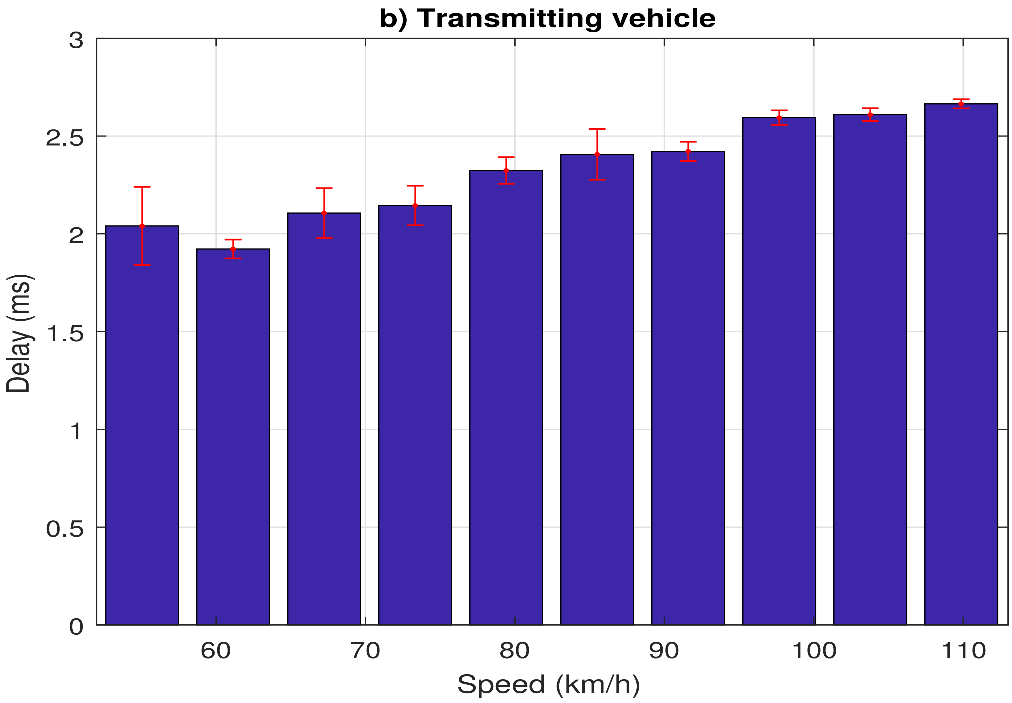

Figure 5.9: Highway scenario - Comparison between packed delay and speed of the transmitting vehicle.

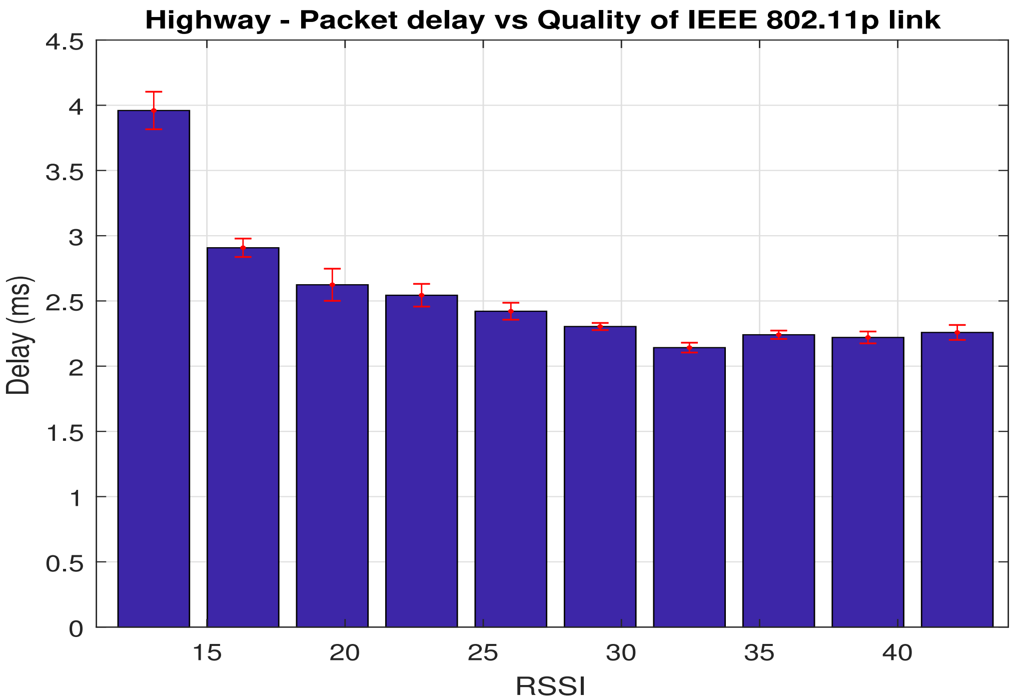

Figure 5.10: Highway scenario - Comparison between packed delay and quality of the IEEE 802.11p link.

Table 5.2: Overall average results for the highway scenario.

Distance (m)	RSSI	Delay (ms)	Bandwidth (Mbps)	Retransmissions (%)
56.90	30.74	2.44	4.71	0.81

When comparing the results of both scenarios we can reach to the following conclusions:

- Communication delay is higher for the highway scenario due to higher distance between vehicles and higher traveling speeds;
- The RSSI is lower on the highway scenario also because of the higher average distance between both vehicles.
- The RSSI does not seem to be a limiting factor, as most of the time the link quality is sufficient for the data involved, even with constant obstructions in the line-of-sight;
- The bandwidth used was the same, which makes sense since the video characteristics were exactly the same in both conditions;
- The packet retransmission rate measured is the same for both scenarios and it is considerably low, being less than 1%.

5.3.3 Considerations

The scenarios tested toke into account that there is only one vehicle transmitting at the same time. However, even considering that the video transmission service is only supposed to be used in very specific situations, in a more realistic environment there could be more videos being transmitted in the network. Furthermore, the data dissemination of both the vehicles and sensory data would be present and as such, needs to have a dedicated amount of bandwidth always available to ensure its correct operation.

Excluding the necessary control data exchanged, the average throughput available on the IEEE 802.11p interface of the devices used was measured to be 11.6 Mbps. The bandwidth used by the data dissemination services was also measured:

- Vehicle data dissemination per vehicle: 5.55 Kbps
- Sensory data dissemination per traffic signal: 0.48 Kbps

Considering a scenario where the data dissemination services running with a total of 50 vehicles and 20 traffic signals, the bandwidth available for the video transmission in the same broadcast range can be easily obtained:

- Total bandwidth available: 11.60 Mbps
- Vehicle data: 277.24 Kbps
- Sensory data: 9.78 Kbps
- Maximum available bandwidth for the V2V video transmission: 11.32 Mbps

In total, the bandwidth used for the video transmission was measured with three different resolutions. Table 5.3 presents the overall maximum availability for the video transmission considering the three different video qualities.

Table 5.3: Video transmission availability

Quality	Bandwidth per Stream (Mbps)	Maximum Available	Remaining Bandwidth (%)
720p @ 25 fps	4.71	2	18.79
480p @ 25 fps	3.10	3	19.83
360p @ 25 fps	1.71	6	11.55

Naturally, the maximum video transmissions available simultaneously is not very high. Nevertheless, a significant part of the bandwidth is still available to be used for other services needed by the passengers on the vehicles, or on occasions where there is more data being disseminated than the scenario considered.

While the main objective of the tests presented was to evaluate whether the WAVE technology was able to support the proposed video transmission system, it is still important to mention that other factors can determine the actual delay between the moment an image is captured and the instant that the same image appears on the receiving vehicle. Most notably, the codification time has a very high impact on this metric when compared to the communication delay. Furthermore, the camera used is connected via WiFi to the CPU, which causes an additional non-negligible delay. For this work this resulted in the following:

- GoPro camera communication delay: 0.80 ms
- Codification delay:
  - 720p: 0.98 s
  - 480p: 0.50 s
  - 360p: 0.24 s

These two limiting factors are independent of the WAVE technology itself and dependent only on the equipment in question. With a more powerful CPU and a camera connected directly via USB, HDMI or Ethernet, these values could be significantly decreased.

5.4 Summary

This chapter presented the tested scenarios and evaluations performed in this dissertation. Despite the inherent subjective characteristic, the data selection mechanisms and display methods developed are presented in a demonstrative manner, with a discussion of their deployment in typical scenarios and the calibrations involved in the information extraction. After this, the tests regarding the V2V video transmission system were presented, with a description of the two scenarios tested and subsequent results. Overall, it can be concluded that the proposed solution for the video transmission between vehicles is able to provide an extra insight for a driver on scenarios with high mobility and velocities with acceptable communication delays.

Chapter 6

Conclusion and Future Work

This chapter presents the overall conclusions of this dissertation as well as the major points that should be improved in the future.

6.1 Conclusion

The main goal of this dissertation was to create an infrastructure capable of collecting contextual data to provide driving assistance. The presented testbed tries to demonstrate the feasibility of the proposed services using well known sensor and communication technologies. The major parts of this work are resumed on the following:

Road Traffic Awareness : This service was created to collect and organize information regarding the position of the neighboring vehicles. The approach proposed makes use of and extends the beacons deployed in the IEEE 802.11p interface to periodically disseminate the GPS coordinates of every vehicle.

Traffic Signals Sensing : A complement for the road traffic awareness, a system that collects sensory data regarding traffic signals was proposed. Using WiFi modules to represent the traffic signals, a vehicle is capable of successfully collecting the data of such signs, either directly via WiFi or via a WAVE broadcast. The use-case proposed in this dissertation only uses information from the traffic signals, however this service is set up in a way that other types of sensory data coming from similar IoT sensors are also supported.

User Feedback : The user feedback is provided using a display screen developed for Android. It is set up to be running an application that is responsible for the interpretation, categorization, storage and display of the data received from the two suggested data collecting services. The data found out to be relevant is displayed on a dynamic map of the neighborhood.

Vehicle-to-Vehicle Video Transmission : Parallel to the rest of the work, a video transmission system between vehicles is also deployed to provide support for assisted overtaking actions. The results show that the time and quality requirements offered by the available technologies are good enough to offer a car-overtaking assistance global service. Two different mobile scenarios, urban and highway, were tested with similar results.

6.2 Future Work

As this dissertation was the starting point for the development of a platform for assisted driving, there are still several important points that need to be addressed:

- Creation of a collision awareness system based on the position, heading and speed of the surrounding vehicles.
- Development of vision-based detection applications using the video images received from neighboring vehicles.
- Evaluation of the proposed data collection services with multiple vehicles and traffic signals simultaneously, in both urban and highway experiments.
- Evaluation of new dissemination strategies for the sensory data, most notably strategies based on the localization of the vehicles.
- Inclusion of other nodes on the overall architecture, such as sensors located on the streets or aerial drones.

Bibliography

[1] Al-Sultan, S., Al-Doori, M. M., Al-Bayatti, A. H., and Zedan, H. (2014). A comprehensive survey on vehicular ad hoc network. *Journal of Network and Computer Applications*, 37(Supplement C):380 – 392.

[2] Alam, N. and Dempster, A. G. (2013). Cooperative positioning for vehicular networks: Facts and future. *IEEE Transactions on Intelligent Transportation Systems*, 14(4):1708–1717.

[3] Ameixieira, C., Matos, J., Moreira, R., Cardote, A., Oliveira, A. S. R., and Sargento, S. (2011). An ieee 802.11p/wave implementation with synchronous channel switching for seamless dual-channel access (poster). *2011 IEEE Vehicular Networking Conference (VNC)*, pages 214–221.

[4] Belyaev, E., Vinel, A., Jonsson, M., and Sjoberg, K. (2014). Live video streaming in ieee 802.11p vehicular networks: Demonstration of an automotive surveillance application. In *2014 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pages 131–132.

[5] Bengler, K., Dietmayer, K., Farber, B., Maurer, M., Stiller, C., and Winner, H. (2014). Three decades of driver assistance systems: Review and future perspectives. *IEEE Intelligent Transportation Systems Magazine*, 6(4):6–22.

[6] Bento, L. C., Bonnifait, P., and Nunes, U. J. (2017). Cooperative gnss positioning aided by road-features measurements. *Transportation Research Part C: Emerging Technologies*, 79(Supplement C):42 – 57.

[7] Bonuccelli, M. A., T., G. G., Lonettit, F., and Martelli, F. (2007). Real-time video transmission in vehicular networks. In *2007 Mobile Networking for Vehicular Environments*, pages 115–120.

[8] C2C:CC (2007). Car 2 car - communication consortium: Manifesto.

[9] Commission, E. (2017). 5g empowering vertical industries.

[10] Diaz-Cacho, M., Pereira, J. F., Delgado, E., and Barreiro, A. (2017). Send-on-delta sampling strategies for vehicle position tracking. In *2017 3rd International Conference on Event-based Control, Communication, and Signal Processing (EBCCSP)*, pages 1–7.

[11] Doshi, A., Cheng, S. Y., and Trivedi, M. M. (2009). A novel active heads-up display for driver assistance. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 39(1):85–93.

[12] FFmpeg Developers (2017). ffmpeg tool (Version 2.8.11) [Software]. `http://ffmpeg.org/`. [Online; accessed 1-November-2017].

[13] Garcia, F., Cerri, P., Broggi, A., de la Escalera, A., and Armingol, J. M. (2012). Data fusion for overtaking vehicle detection based on radar and optical flow. In *2012 IEEE Intelligent Vehicles Symposium*, pages 494–499.

[14] Gomes, P., Olaverri-Monreal, C., and Ferreira, M. (2012). Making vehicles transparent through v2v video streaming. *IEEE Transactions on Intelligent Transportation Systems*, 13(2):930–938.

[15] Hafner, M. R., Cunningham, D., Caminiti, L., and Vecchio, D. D. (2013). Cooperative collision avoidance at intersections: Algorithms and experiments. *IEEE Transactions on Intelligent Transportation Systems*, 14(3):1162–1175.

[16] Hameed Mir, Z. and Filali, F. (2014). Lte and ieee 802.11p for vehicular networking: a performance evaluation. *EURASIP Journal on Wireless Communications and Networking*, 2014(1):89.

[17] Hartenstein, H. and Laberteaux, L. P. (2008). A tutorial survey on vehicular ad hoc networks. *IEEE Communications Magazine*, 46(6):164–171.

[18] Hultqvist, D., Roll, J., Svensson, F., Dahlin, J., and Schn, T. B. (2014). Detecting and positioning overtaking vehicles using 1d optical flow. In *2014 IEEE Intelligent Vehicles Symposium Proceedings*, pages 861–866.

[19] IEEE (2006a). IEEE trial-use standard for wireless access in vehicular environments - security services for applications and management messages. *IEEE Std 1609.2-2006*, pages 0_1–105.

[20] IEEE (2006b). IEEE trial-use standard for wireless access in vehicular environments (wave) - multi-channel operation. *IEEE Std 1609.4-2006 -Test*, pages 1–82.

[21] IEEE (2006c). Trial-use standard for wireless access in vehicular environments (wave) - resource manager. *IEEE Std 1609.1-2006*, pages 1–71.

[22] IEEE (2010a). IEEE standard for information technology– local and metropolitan area networks– specific requirements– part 11: Wireless lan medium access control (mac) and physical layer (phy) specifications amendment 6: Wireless access in vehicular environments. *IEEE Std 802.11p-2010 (Amendment to IEEE Std 802.11-2007 as amended by IEEE Std 802.11k-2008, IEEE Std 802.11r-2008, IEEE Std 802.11y-2008, IEEE Std 802.11n-2009, and IEEE Std 802.11w-2009)*, pages 1–51.

[23] IEEE (2010b). IEEE standard for wireless access in vehicular environments (wave) - networking services. *IEEE Std 1609.3-2010 (Revision of IEEE Std 1609.3-2007)*, pages 1–144.

[24] IEEE (2011). IEEE standard for wireless access in vehicular environments (wave)–multi-channel operation. *IEEE Std 1609.4-2010 (Revision of IEEE Std 1609.4-2006)*, pages 1–89.

[25] Jiang, D. and Delgrossi, L. (2008). Ieee 802.11p: Towards an international standard for wireless access in vehicular environments. In *VTC Spring 2008 - IEEE Vehicular Technology Conference*, pages 2036–2040.

[26] Khairnar, V. D. and Pradhan, S. N. (2013). Comparative study of simulation for vehicular ad-hoc network. *CoRR*, abs/1304.5181.

[27] Liu, J., Wan, J., Wang, Q., Deng, P., Zhou, K., and Qiao, Y. (2016). A survey on position-based routing for vehicular ad hoc networks. *Telecommunication Systems*, 62(1):15–30.

[28] McCall, J. C. and Trivedi, M. M. (2006). Video-based lane estimation and tracking for driver assistance: survey, system, and evaluation. *IEEE Transactions on Intelligent Transportation Systems*, 7(1):20–37.

[29] Mogelmose, A., Trivedi, M. M., and Moeslund, T. B. (2012). Vision-based traffic sign detection and analysis for intelligent driver assistance systems: Perspectives and survey. *IEEE Transactions on Intelligent Transportation Systems*, 13(4):1484–1497.

[30] Morgan, Y. L. (2010). Notes on dsrc & wave standards suite: Its architecture, design, and characteristics. *IEEE Communications Surveys & Tutorials*, 12(4):504–518.

[31] Moustafa, H. and Zhang, Y. (2009). *Vehicular Networks: Techniques, Standards, and Applications*. Auerbach Publications, Boston, MA, USA, 1st edition.

[32] Mukhtar, A., Xia, L., and Tang, T. B. (2015). Vehicle detection techniques for collision avoidance systems: A review. *IEEE Transactions on Intelligent Transportation Systems*, 16(5):2318–2338.

[33] Mgelmose, A., Liu, D., and Trivedi, M. M. (2014). Traffic sign detection for u.s. roads: Remaining challenges and a case for tracking. In *17th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, pages 1394–1399.

[34] NEKOVEE, M. (2005). Sensor networks on the road : the promises and challenges of vehicular adhoc networks and vehicular grids. *Proc. Workshop on Ubiquitous Computing and e-Research, May 2005.*

[35] Park, J.-S., Lee, U., Oh, S. Y., Gerla, M., and Lun, D. S. (2006). Emergency related video streaming in vanet using network coding. In *Proceedings of the 3rd international workshop on Vehicular ad hoc networks*, pages 102–103. ACM.

[36] Ramirez, A., Ohn-Bar, E., and Trivedi, M. (2014). Integrating motion and appearance for overtaking vehicle detection. In *2014 IEEE Intelligent Vehicles Symposium Proceedings*, pages 96–101.

[37] Rozas-Ramallal, I., Fernández-Caramés, T. M., Dapena, A., and García-Naya, J. A. (2013). Evaluation of h.264/avc over ieee 802.11p vehicular networks. *EURASIP Journal on Advances in Signal Processing*, 2013(1):77.

[38] Schoch, E., Kargl, F., and Weber, M. (2008). Communication patterns in vanets. *IEEE Communications Magazine*, 46(11):119–125.

[39] Sivaraman, S. and Trivedi, M. M. (2013). Looking at vehicles on the road: A survey of vision-based vehicle detection, tracking, and behavior analysis. *IEEE Transactions on Intelligent Transportation Systems*, 14(4):1773–1795.

[40] Soldo, F., Casetti, C., Chiasserini, C. F., and Chaparro, P. A. (2011). Video streaming distribution in vanets. *IEEE Transactions on Parallel and Distributed Systems*, 22(7):1085–1091.

[41] Sun, Z., Bebis, G., and Miller, R. (2006). On-road vehicle detection: a review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(5):694–711.

[42] Tan, H. S. and Huang, J. (2006). Dgps-based vehicle-to-vehicle cooperative collision warning: Engineering feasibility viewpoints. *IEEE Transactions on Intelligent Transportation Systems*, 7(4):415–428.

[43] Tatchikou, R., Biswas, S., and Dion, F. (2005). Cooperative vehicle collision avoidance using inter-vehicle packet forwarding. In *GLOBECOM '05. IEEE Global Telecommunications Conference, 2005.*, volume 5, pages 5 pp.–2766.

[44] TNO (2007). 2007-d-r0048/b report.

[45] Venkataraman, H., d'Ussel, A., Corre, T., Muntean, C. H., and Muntean, G.-M. (2010). Performance analysis of real-time multimedia transmission in 802.11p based multihop hybrid vehicular networks. In *Proceedings of the 6th International Wireless Communications and Mobile Computing Conference*, IWCMC '10, pages 1151–1155, New York, NY, USA. ACM.

[46] Vinel, A., Belyaev, E., Egiazarian, K., and Koucheryavy, Y. (2012). An overtaking assistance system based on joint beaconing and real-time video transmission. *IEEE Transactions on Vehicular Technology*, 61(5):2319–2329.

[47] Vinel, A., Belyaev, E., Lamotte, O., Gabbouj, M., Koucheryavy, Y., and Egiazarian, K. (2013). Video transmission over ieee 802.11p: Real-world measurements. In *2013 IEEE International Conference on Communications Workshops (ICC)*, pages 505–509.

[48] Wen, X., Shao, L., Xue, Y., and Fang, W. (2015). A rapid learning algorithm for vehicle classification. *Information Sciences*, 295(Supplement C):395 – 406.

[49] WHO (2015). Global status report on road safety 2015. `http://www.who.int/violence_injury_prev-\ention/road_safety_status/2015/en/`. Accessed: 07-09-2017.

[50] William J. Hughes Technical Center. WAAS T&E Team (2017). Global Positioning System (GPS), Standard Positioning Service (SPS), Performance Analysis Report. Technical report, Federal Aviation Administration. GPS Product Team, 1284 Maryland Avenue SW. Washington, DC 20024. United States of America.

[51] Willke, T. L., Tientrakool, P., and Maxemchuk, N. F. (2009). A survey of inter-vehicle communication protocols and their applications. *IEEE Communications Surveys Tutorials*, 11(2):3–20.

[52] Yang, X., Liu, L., Vaidya, N. H., and Zhao, F. (2004). A vehicle-to-vehicle communication protocol for cooperative collision warning. In *The First Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services, 2004. MOBIQUITOUS 2004.*, pages 114–123.

[53] Yousefi, S., Mousavi, M. S., and Fathy, M. (2006). Vehicular ad hoc networks (vanets): Challenges and perspectives. In *2006 6th International Conference on ITS Telecommunications*, pages 761–766.

[54] Zardosht, B., Beauchemin, S., and Bauer, M. A. (2013). A decision making module for cooperative collision warning systems using vehicular ad-hoc networks. In *16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013)*, pages 1743–1749.

[55] Zeadally, S., Hunt, R., Chen, Y.-S., Irwin, A., and Hassan, A. (2012). Vehicular ad hoc networks (vanets): status, results, andchallenges. *Telecommunication Systems*, 50(4):217–241.